Detecting Quantum Insert
Using Bro-IDS

5 August - BroCon 2015
Yun Zheng Hu
Fox-IT Security Research Team
$ whoami

Yun Zheng Hu
Principal Security Expert

www.fox-it.com
github.com/fox-it
@YunZhengHu
Fox-IT
Delft, Netherlands
Past contributions to Bro

- **BIT-968**: bytestring_to_count()
- **BIT-969**: reverse()
Agenda

• What is QUANTUM INSERT?
• How to perform QUANTUM INSERT?
• Detection
• Demo
• Injections we detected in the wild
What is QUANTUMINSERT?
What is QUANTUMINSERT?

- Snowden leaks
- Codename for TCP hijacking
  - Specifically targeting HTTP
  - More injection than hijacking
- React faster than other servers
  - Win race condition
# Other QUANTUM attacks

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUANTUMDNS</td>
<td>DNS Injection/Redirection of A records</td>
</tr>
<tr>
<td>QUANTUMBOT</td>
<td>Hijacking idle IRC bots and c&amp;c communication from bots.</td>
</tr>
<tr>
<td>QUANTUMSKY</td>
<td>Deny access to webpage by injecting/spoofing RST packets</td>
</tr>
<tr>
<td>QUANTUMBISCUIT</td>
<td>Enhance QI behind large proxies</td>
</tr>
</tbody>
</table>

source: [https://firstlook.org/theintercept/document/2014/03/12/one-way-quantum/](https://firstlook.org/theintercept/document/2014/03/12/one-way-quantum/)
Slide that started it all

Heuristic Example

- QUANTUM
  - It's no lie, quantum is cool.
    - But its easy to find
  - Analyze first content carrying packet
    - Check for sequence number duplication, but different data size
    - If content differs within the first 10% of the pkt payload, alert.
Security Research Team

- How does it really work?
  - Perform a successful Quantum Insert
  - Capture a PCAP (or it didn’t happen)
  - Check existing IDS software for detection
Detect "quantum insert" type of attacks

Add detection for "quantum insert" type of attacks. Since the leaked information is classified, I will try to explain in unclassified form what it is about.

The idea is that you have a passive adversary that sniff your TCP sequence numbers and inject its malicious payload faster than the real server.

One of the leaked documents mentions as an alerting mechanism to detect duplicate TCP sequence numbers from same source, where at least 10% of the beginning of the content of the two packets differs.

Jon Siwek added a comment - 09/Feb/15 9:29 AM

Handling the "rexmit_inconsistency" event and comparing the mismatched content might be a way to do what you want.

Initial IDS Coverage

- Bro should detect it using `rexmit_inconsistency`, but it didn’t work
- Snort protocol decoders did not trigger anything
- Suricata did not trigger anything, needed:
  - `stream-event:reassembly_overlap不同的data`
Howto QUANTUM
TCP 3-way Handshake

1. **SYN**
2. **SYN/ACK response**
3. **ACK**
TCP Hijacking

- Kevin Mitnick
  - Successfully hijacked a remote TCP session
  - Predicted the TCP sequence numbers
  - Nowadays, TCP sequence numbers are random
  - Have to sniff and leak the information
QI vs TCP Injection

- Quantum Insert is TCP packet injection
- But specifically against HTTP sessions
- Confirms target by checking tracking Cookies
- Uses a **monitor** to leak the information
- Uses a **shooter** to spoof and insert the packet
Requirements

• Observe & Leak TCP Session information
• Able to spoof packets
• Racing the response (be faster)
TCP Injection

- Client
  - SYN

- Router
  - Skull

- Shooter

- Server
TCP Injection

Client -> Router (seq=x) -> Shooter (SYN) -> Server
TCP Injection

Client → Router (seq=x) → Shooter → Server

- SYN
- SYN+ACK
TCP Injection

Client

Router

Shooter

Server

SYN

seq=x

ack=x+1, seq=y
TCP Injection

Client

SYN+ACK
ACK

Router

seq=x

ack=x+1, seq=y

Shooter

Server

SYN
TCP Injection

Client → Router → Shooter → Server

SYN+ACK

seq=x

ack=x+1, seq=y

ack=y+1, seq=x+1

SYN

ACK
TCP Injection

Client

SYN+ACK

PSH+ACK HTTP GET

Router

seq=x

ack=x+1, seq=y

ack=y+1, seq=x+1

Server

SYN

ACK

Shooter
TCP Injection

Client

SYN+ACK

Router

seq=x

ack=x+1, seq=y

ack=y+1, seq=x+1

ack=y, seq=x

Shooter

Server

SYN

ACK

PSH+ACK

HTTP GET

QI TIP

{src,dst} {ip,port}

x, y, len
TCP Injection

Client

SYN+ACK

Router

seq=x

ack=x+1, seq=y

ack=y+1, seq=x+1

ack=y, seq=x

Shooter

Server

SYN

ACK

PSH+ACK

HTTP GET

PSH+ACK

302 Redirect

ACK

QI TIP

{src, dst} {ip, port} x, y, len

PSH+ACK

302 Redirect

ACK
TCP Injection

**Client**
- SYN+ACK

**Router**
- seq=x
- ack=x+1, seq=y
- ack=y+1, seq=x+1
- ack=y, seq=x

**Shooter**
- QI TIP
  - {src,dst} {ip,port}
  - x, y, len

**Server**
- SYN
- ACK
- PSH+ACK
- HTTP GET
- ACK

**QI TIP**
- {src, dst} {ip, port}
- x, y, len

**Grafik**
- Firefox
- Router
- Shooter
- Server

**HTTP GET**
- ack=x+len, seq=y

**302 Redirect**
- ack=x+len, seq=y
TCP Injection

Client

SYN+ACK

PSH+ACK

302 Redirect

ACK

Router

seq=x

ack=x+1, seq=y

ack=y+1, seq=x+1

ack=y, seq=x

Shooter

Server

SYN

ACK

PSH+ACK

HTTP GET

QI TIP

{src,dst} {ip,port}

x, y, len

ack=x+len, seq=y

ack=x+len, seq=y
TCP Injection

Client
SYN+ACK
PSH+ACK
302 Redirect
ACK

Router
seq=x
ack=x+1, seq=y
ack=y+1, seq=x+1
ack=y, seq=x

Shooter

Server
SYN
ACK
PSH+ACK
HTTP GET

QI TIP
{src,dst} {ip,port} x, y, len

ACK
ack=x+len, seq=y

ACK
ack=x+len, seq=y

PSH+ACK
200 OK
TCP Injection

Client -> Router -> Shooter -> Server

Client sends SYN+ACK, seq=x

Server sends SYN, seq=x

Client sends ACK, ack=x+1, seq=y

Server sends SYN+ACK, ack=x+1, seq=y

Client sends ACK, ack=y+1, seq=x+1

Server sends ACK, ack=y, seq=x

QI TIP

{x, y, len}

PSH+ACK, 302 Redirect

PSH+ACK, 200 OK
TCP segment overlap

- Client receives:
  - Spoofed & Inserted packet
  - Original HTTP response packet
- Attacker can easily solve this, eg by specifying:
  - Content-Length: 0
Overlapping TCP segments

HTTP/1.1 302 Found
Location: http://fox-it.com/
Content-Length: 0

Packet #1 - Sequence 1 (Length 71)
Overlapping TCP segments

HTTP/1.1 302 Found
Location: http://fox-it.com/
Content-Length: 0

Packet #1 - Sequence 1 (Length 71)

Last-Modified: Tue, 21 Apr 2015 19:16:41 GMT
Connection: close
ETag: "5536a219-1caf5"
Accept-Ranges: bytes
Vary: Accept-Encoding, User-Agent
Content-Encoding: gzip
Transfer-Encoding: chunked

6dca ...

Packet #2 - Sequence 1 - (Length 1448)
Overlapping TCP segments

HTTP/1.1 302 Found
Location: http://fox-it.com/
Content-Length: 0

Last-Modified: Tue, 21 Apr 2015 19:16:41 GMT
Connection: close
ETag: "5536a219-1caf5"
Accept-Ranges: bytes
Vary: Accept-Encoding, User-Agent
Content-Encoding: gzip
Transfer-Encoding: chunked

6dca ...
Getting more speed

• Injecting on the first SYN-ACK response from the Server

• Improved speed

• But cannot confirm request/victim
Detecting Quantum Insert
How to detect QI

• QI results in duplicate sequence numbers
• Which means TCP segment overlap
• Check if overlapping segments are different
Other packet artefacts

- Time to Live usually differs from other packets
- Can give away where in the chain the packets are being injected
- Could have different TCP options
Bro policy

- Uses tcp_packet callback
- keeps track of the last sequence number and payload of a connection
- check for duplicate sequence numbers
- check for payload difference
- Inefficient but works
Bro patches

- Integrated in the TCP Reassembly code
- Rolling buffer of old segments, configureable using `tcp_max_old_segments`
- Overlapping segments with different data will trigger the `rexmit_inconsistency` event
- Merged in commit `c1f060be` on June 28 2015
Demo
Demo Setup

LAN
- Target
- Shooter

WAN
- Internet

Router
TCP Injections in the wild
Examples of detected QI

- Network Appliances performing TCP injection
  - Blocking content, such as ads
- Some Chinese websites result in TCP injection
  - Mostly for blocking purposes
False positives?

- SSL Traffic
- Window size changes
- Recommendations:
  - Ignore SSL/TLS
  - Limit to HTTP responses
Research

• All the research, pcaps, and tools are published on our GitHub and blog:
  
  • https://github.com/fox-it/quantuminsert
  
  • blog.fox-it.com/2015/04/20/deep-dive-into-quantum-insert/
Recommendations

• As a server
  • Use SSL + HTTP Strict Transport Security
  • Resources should be over SSL as well
• As a client
  • Use https directly, don’t rely on redirects
  • Isolated VM for browsing only
Questions?

FOX IT
FOR A MORE SECURE SOCIETY
Bonus Bro policy!

- meterpreter.bro
  - Detect Metasploit meterpreter payload transfer
  - Nice for lateral movement detection!
  - Uses sequence numbers to check the size

- Will be available after the talk:
  - https://github.com/fox-it/bro-scripts