The Bro Debugger

Vlad Grigorescu
NCSA
> whoami

Member of the Bro development team

Security Engineer at the National Center for Supercomputing Applications (NCSA)

https://github.com/grigorescu

@0f010d
function send_reply(c: connection, msg: dns)
{
    # TODO: Delete this
    print "In send_reply";
    print "c":", c;
    print "dns_msg":", dns_msg;
    print "query":", query;
    print "qtype":", qtype;
    print "qclass":", qclass;

    if ( qtype != 1 )
    {
        # Only A query types are supported
        return;
    }

    print "Still in send_reply...";

    if ( qclass != 1 )
    {
        # Only IN query classes are supported
        return;
    }

    print "I'm not quite dead yet...";
There's a better way...
bro --debug-policy
"GDB for Bro Scripts"

- Debugger for script-land
- No visibility into the "core layer" (C/C++ code)
- Breakpoints, flow control, examining values
- Executing Bro statements
- Can even be used on live traffic (not recommended)
Breakpoints

• Set breakpoints at script locations

```c
event connection_state_remove(c: connection)
{
    local ssh_servers = [ 10.2.4.1, 10.2.4.2 ];

    if ( c$id$resp_p == 22/tcp )
    {
        if ( c$id$resp_h !in ssh_servers )
            print string_cat("Alert! SSH server: ", c$id$resp_h);
    }
}
```
Breakpoints
## Breakpoints

<table>
<thead>
<tr>
<th>Command</th>
<th>Breakpoint at:</th>
</tr>
</thead>
<tbody>
<tr>
<td>break</td>
<td>Current location</td>
</tr>
<tr>
<td>break 3</td>
<td>Line 3 of current file</td>
</tr>
<tr>
<td>break error1.bro:3</td>
<td>Line 3 of error1.bro</td>
</tr>
<tr>
<td>break bro_init</td>
<td>bro_init function/event</td>
</tr>
<tr>
<td>break irc_*</td>
<td>irc_* function/events</td>
</tr>
</tbody>
</table>
# Breakpoints

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>info breakpoints</td>
<td>Show list of breakpoints</td>
</tr>
<tr>
<td>enable 1</td>
<td>Enable breakpoint #1</td>
</tr>
<tr>
<td>disable 1</td>
<td>Disable breakpoint #1</td>
</tr>
<tr>
<td>delete 1</td>
<td>Delete breakpoint #1</td>
</tr>
<tr>
<td>continue (c)</td>
<td>Resume execution</td>
</tr>
<tr>
<td>C-c</td>
<td>Stop execution</td>
</tr>
</tbody>
</table>
Examining State
# Examining State

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Show up to 10 lines of code</td>
</tr>
<tr>
<td>list 3</td>
<td>Show ±5 lines around line 3</td>
</tr>
<tr>
<td>list error1.bro:3</td>
<td>...around error1.bro:3</td>
</tr>
<tr>
<td>list bro_init</td>
<td>...around the bro_init event</td>
</tr>
<tr>
<td>print $exp (p)</td>
<td>Evaluate and print $exp</td>
</tr>
</tbody>
</table>
Flow Control

```python
event connection_state_remove(c: connection)
{
    local ssh_servers = [ 10.2.4.1, 10.2.4.2 ];

    if ( c$id$resp_p == 22/tcp )
    {
        if ( c$id$resp_h !in ssh_servers )
            print string_cat("Alert! SSH server: ", c$id$resp_h);
    }
}
```

```
vladg ~ src > bro > tmp bro -r ssh_sample.pcap error1.bro
1445023194.281899 fatal error in <no location>: Val::CONST_ACCESSOR (addr/string) (141.142.112.248)
```
# Flow Control

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cond 1 c$?id</td>
<td>Add condition to breakpoint 1</td>
</tr>
<tr>
<td>next (n)</td>
<td>Next line, don't enter funcs</td>
</tr>
<tr>
<td>step</td>
<td>Next line, do enter funcs</td>
</tr>
<tr>
<td>finish</td>
<td>Run until end of current func</td>
</tr>
</tbody>
</table>
Extra Credit

- Setting condition breakpoints can be very powerful
- syslog(string)
- system(command)
- dump_current_packet(file_name)
breakpoint_to_pcap.sh

• Can filter a PCAP file
• Filters all connections that hit a certain point in the code
• Can pinpoint traffic that causes protocol errors, weirds, crashes, etc.
breakpoint_to_pcap.sh

```bash
$vldg ➜ src ➜ bro ➜ tmp$ tshark -r ~/pcaps/dns_multicast_bug.pcap | wc -l


Policy file debugging ON.
In bro_init() at /Users/vldg/src/bro/scripts/base/frameworks/sumstats/.main.bro:276
276 hook register_observe_plugins();
Setting breakpoint on /Users/vldg/src/bro/scripts/base/protocols/dns/.main.bro:414:
Breakpoint set at:
for ( i in strs )
{
    if ( i > 0 )
        txt_strings += " ";
    txt_strings += fmt("TXT %d %s", |strs[i]|, strs[i]);
}
Continuing.
(Bro [8]) (Bro [1]) (Bro [2]) (Bro [3]) (Bro [4]) %
$vldg ➜ src ➜ bro ➜ tmp$ tshark -r test_output.pcap | wc -l
86
$vldg ➜ src ➜ bro ➜ tmp$
```

http://go.ncsa.illinois.edu/breakpoint_to_pcap