Brute forcing in the Shadows
Evading Automated Detection

Martin Drašar, Jan Vykopal
{drasar|vykopal}@ics.muni.cz

Institute of Computer Science
Masaryk University
Brno, Czech Republic

FloCon 2012
January 12, Austin, Texas
Part I

Network Security Monitoring at Masaryk University, Brno
Masaryk University, Brno

- **2nd largest** university in the Czech Republic.
- ~45,000 students, ~5,000 staff.
- ~15,000 **hosts** (public IPs) online every day.

CSIRT-MU

- In charge of university network security since 2009.
- Accredited by European Trusted Introducer in 2011.
- ~3 FTE ⇒ **automatization**.
- **Intrusion detection** is one of key services.
NetFlow-based

- ~35 software (1 GE) or hardware accelerated (10 GE) stand-alone probes at important links.
- Several NfSen collectors for data storage.
- Wealth of in-house developed detection tools.

Other

- Low- and high-interaction honeypots.
- Integrated logging system based on Syslog.
- External data sources: Shadowserver and Team Cymru reports.
Intrusion Detection at Masaryk University contd.

FlowMon probe

NetFlow data acquisition

NetFlow collector

NetFlow v5/v9

SPAM detection

Worm/virus detection

intrusion detection

incident reporting

1/10 GE

EWS

http

WWW

mail

syslog

mailbox

syslog server

Drasar, Vykopal

Bruteforcing in the Shadows – Evading Automated Detection
In-house Developed Tools

Our tools detects:

- live hosts and missing DNS reverse entries,
- port scans,
- embedded botnets (presented at FloCon 2011),
- network address translators,
- spamming hosts,
- changes in RTT of selected servers,
- **bruteforce attacks**.

In progress: analytic tool of phishing incidents.
Part II

Bruteforcing: State of the Art
Bruteforce SSH Attacks: Introduction

- Repetitive online password guessing.
- Ubiquitous on the Internet.
- Humans tend to select weak passwords.
- Still a threat (stepping stones, data leakage, ...).
- Common attack types – 1:1, 1:N, M:1, and M:N.

- Traditionally detected at the host level (1:1 and M:1).
- Host-based detection may miss large attacks (1:N and M:N).
- Network-based detection promises to protect devices that do not protect themselves (such as embedded devices, routers).
Detection methods are developed using various datasets, but academia uses data sets that do not reflect the reality.

- Datasets do not contain current attacks ⇒ old.
- Data are from limited topologies and small number of networked hosts ⇒ small-scaled.
- They are not (sufficiently) annotated, leave you on your own.
- Favor volume-visible attacks such as scans, floods, DoS, etc.
- May or may not contain stealth attacks.
- Security research bent on detecting attacks present in datasets only.

We prefer real examples from daily life of CSIRT-MU.
Typical brute force SSH attack scenario:

- Uses common SSH port.
- Repetitive password guessing generates *similar* flows in terms of volume and time.
- Small and short flows indicates unsuccessful attempts.
- It is often preceded by port scanning.
- One attacker aims at several (many) hosts.
- Attacks continue even after black-holing.

We believe it is the similar case for other services/protocols: Telnet, RDP, web authentication, etc.
Increasing Effectiveness of Detection

- Know your network. Is it server or client?
- Have the attackers scanned our network already?
- Do they access honeypots?
- Are they from untrusted autonomous system?
- Use external data sources but be careful!
Part III

Evading Automated Detection
Typical bruteforce attack [THC-Hydra, ...]

- Attacking as much as it is possible.
- Using predefined attacks per time frame.

Sophisticated bruteforce attack [Ncrack]

- Adaptive lowering of attack attempts per time frame to get under network thresholds.
- Threshold detection is not that hard.
  - Especially when there is an imminent blocking of attackers.
- Possible if attacker controls more hosts.
Random Delays

Typical bruteforce attack [THC-Hydra, ...]

- Fixed attempts per time frame.
- Zero or fixed interval between attempts.

Sophisticated bruteforce attack [Ncrack]

- Simulation of real traffic by inserting random delays between attack attempts.
- Random delays cause variations in the number of attempts per timeframe.
Random Delays: Illustration

**Time invariant attacks [8 parallel, 1 s delay]**

Flows

4 second time windows

1  2  3  4  5  6

green ~ blue ~ red
Random Delays: Illustration contd.

Time variant attacks [8 parallel, 1–13 s delay]

Flows

10 second time windows

green !~ blue !~ red
Flow Stretching

Typical bruteforce attack [THC-Hydra, Ncrack, ...]

- Exchanging bare minimum of data needed for authentication attempt.

Sophisticated bruteforce attack [?]

- Flow-wise the volume and duration are the only difference between successful and failed authentication.
- Exploiting protocol design to mimic successful authentication by increasing volume and duration.
- SSH, RDP, HTTP and probably others.
Flow Stretching: Illustration

Non-stretched flows

<table>
<thead>
<tr>
<th>Duration</th>
<th>Protocol</th>
<th>Src IP:Src Port</th>
<th>Dst IP:Port</th>
<th>Packets</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.310</td>
<td>TCP</td>
<td>147.251.AA.BB:49297 -&gt; 147.251.CC.DD:22</td>
<td>12</td>
<td>1197</td>
<td></td>
</tr>
<tr>
<td>0.269</td>
<td>TCP</td>
<td>147.251.AA.BB:49320 -&gt; 147.251.CC.DD:22</td>
<td>11</td>
<td>1157</td>
<td></td>
</tr>
<tr>
<td>0.436</td>
<td>TCP</td>
<td>147.251.AA.BB:49329 -&gt; 147.251.CC.DD:22</td>
<td>11</td>
<td>1157</td>
<td></td>
</tr>
<tr>
<td>0.196</td>
<td>TCP</td>
<td>147.251.AA.BB:49358 -&gt; 147.251.CC.DD:22</td>
<td>11</td>
<td>1173</td>
<td></td>
</tr>
<tr>
<td>0.155</td>
<td>TCP</td>
<td>147.251.AA.BB:49308 -&gt; 147.251.CC.DD:22</td>
<td>11</td>
<td>1157</td>
<td></td>
</tr>
<tr>
<td>0.273</td>
<td>TCP</td>
<td>147.251.AA.BB:49318 -&gt; 147.251.CC.DD:22</td>
<td>11</td>
<td>1157</td>
<td></td>
</tr>
<tr>
<td>0.270</td>
<td>TCP</td>
<td>147.251.AA.BB:49343 -&gt; 147.251.CC.DD:22</td>
<td>11</td>
<td>1157</td>
<td></td>
</tr>
<tr>
<td>0.259</td>
<td>TCP</td>
<td>147.251.AA.BB:49344 -&gt; 147.251.CC.DD:22</td>
<td>11</td>
<td>1157</td>
<td></td>
</tr>
<tr>
<td>0.206</td>
<td>TCP</td>
<td>147.251.AA.BB:49355 -&gt; 147.251.CC.DD:22</td>
<td>11</td>
<td>1173</td>
<td></td>
</tr>
<tr>
<td>0.190</td>
<td>TCP</td>
<td>147.251.AA.BB:49362 -&gt; 147.251.CC.DD:22</td>
<td>11</td>
<td>1157</td>
<td></td>
</tr>
</tbody>
</table>
### Stretched flows

<table>
<thead>
<tr>
<th>Duration</th>
<th>Protocol</th>
<th>Src IP:Src Port</th>
<th>Dst IP:Port</th>
<th>Packets</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.157</td>
<td>TCP</td>
<td>147.251.AA.BB:49368 -&gt; 147.251.CC.DD:22</td>
<td>142</td>
<td>44441</td>
<td></td>
</tr>
<tr>
<td>5.501</td>
<td>TCP</td>
<td>147.251.AA.BB:49379 -&gt; 147.251.CC.DD:22</td>
<td>99</td>
<td>30389</td>
<td></td>
</tr>
<tr>
<td>14.227</td>
<td>TCP</td>
<td>147.251.AA.BB:49367 -&gt; 147.251.CC.DD:22</td>
<td>239</td>
<td>76837</td>
<td></td>
</tr>
<tr>
<td>6.722</td>
<td>TCP</td>
<td>147.251.AA.BB:49369 -&gt; 147.251.CC.DD:22</td>
<td>119</td>
<td>36981</td>
<td></td>
</tr>
<tr>
<td>5.429</td>
<td>TCP</td>
<td>147.251.AA.BB:49369 -&gt; 147.251.CC.DD:22</td>
<td>98</td>
<td>9865</td>
<td></td>
</tr>
<tr>
<td>18.184</td>
<td>TCP</td>
<td>147.251.AA.BB:49375 -&gt; 147.251.CC.DD:22</td>
<td>302</td>
<td>97593</td>
<td></td>
</tr>
<tr>
<td>2.239</td>
<td>TCP</td>
<td>147.251.AA.BB:49387 -&gt; 147.251.CC.DD:22</td>
<td>47</td>
<td>13125</td>
<td></td>
</tr>
<tr>
<td>1.304</td>
<td>TCP</td>
<td>147.251.AA.BB:49380 -&gt; 147.251.CC.DD:22</td>
<td>32</td>
<td>8033</td>
<td></td>
</tr>
<tr>
<td>23.320</td>
<td>TCP</td>
<td>147.251.AA.BB:49374 -&gt; 147.251.CC.DD:22</td>
<td>384</td>
<td>124865</td>
<td></td>
</tr>
<tr>
<td>1.798</td>
<td>TCP</td>
<td>147.251.AA.BB:49386 -&gt; 147.251.CC.DD:22</td>
<td>40</td>
<td>10737</td>
<td></td>
</tr>
</tbody>
</table>
Search for a Remedy

- Correlation with (sys)log information?
- Bigger restrictiveness?
- More sophisticated flow analyses?
- **Open problem...**
- Brute-force (SSH) attacks are still a threat.
- It is possible to detect them by NetFlow analysis.
- Defenders are prepared for simple attacks and simple attackers.
- Many NetFlow detection methods can be easily evaded.
- Limitations of NetFlow – how to overcome them?
Bruteforcing in the Shadows
Evading Automated Detection

This material is based upon work supported by the
Czech Ministry of Defence under Contract No. OVMASUN200801.