RECON FOR THE DEFENDER: YOU KNOW NOTHING (ABOUT YOUR ASSETS)

Ed Bellis
CTO, Co-founder
Kenna Security
@ebellis

Jonathan Cran
Head of Research
Kenna Security
@jcran
About Your Presenters

Ed Bellis, CTO & Founder

Founded Kenna security in 2010 to help organizations get a true picture of risk. Formerly... CISO, Orbitz, Bank of America.

Jonathan Cran, Head of Research

Recovering penetration tester. Formerly... Bugcrowd, Rapid7, Metasploit. Also, creator of Intrigue discovery framework.
Part I: The Case for Recon: Challenges of real-world asset and vulnerability discovery

Part 2: Adversarial Perspective: What techniques can we utilize from attackers

Part 3: Integrating Recon Techniques: Affecting your Risk Management program
The Need for Visibility

<table>
<thead>
<tr>
<th>CVE-2017-5638</th>
<th>Latest</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>Total</th>
<th>Std Dev</th>
<th>95%</th>
<th>99%</th>
</tr>
</thead>
<tbody>
<tr>
<td>November</td>
<td>31,885</td>
<td>46.0</td>
<td>58,671</td>
<td>8,157</td>
<td>1,435,645</td>
<td>10,938</td>
<td>6,549 – 9,765</td>
<td>6,034 – 10.9k</td>
</tr>
<tr>
<td>December</td>
<td>31,885</td>
<td>46.0</td>
<td>58,671</td>
<td>8,157</td>
<td>1,435,645</td>
<td>10,938</td>
<td>6,549 – 9,765</td>
<td>6,034 – 10.9k</td>
</tr>
<tr>
<td>January</td>
<td>31,885</td>
<td>46.0</td>
<td>58,671</td>
<td>8,157</td>
<td>1,435,645</td>
<td>10,938</td>
<td>6,549 – 9,765</td>
<td>6,034 – 10.9k</td>
</tr>
<tr>
<td>February</td>
<td>31,885</td>
<td>46.0</td>
<td>58,671</td>
<td>8,157</td>
<td>1,435,645</td>
<td>10,938</td>
<td>6,549 – 9,765</td>
<td>6,034 – 10.9k</td>
</tr>
<tr>
<td>March</td>
<td>31,885</td>
<td>46.0</td>
<td>58,671</td>
<td>8,157</td>
<td>1,435,645</td>
<td>10,938</td>
<td>6,549 – 9,765</td>
<td>6,034 – 10.9k</td>
</tr>
<tr>
<td>April</td>
<td>31,885</td>
<td>46.0</td>
<td>58,671</td>
<td>8,157</td>
<td>1,435,645</td>
<td>10,938</td>
<td>6,549 – 9,765</td>
<td>6,034 – 10.9k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>latest</th>
<th>min</th>
<th>max</th>
<th>mean</th>
<th>total</th>
<th>std dev</th>
<th>95%</th>
<th>99%</th>
</tr>
</thead>
<tbody>
<tr>
<td>31,885</td>
<td>46.0</td>
<td>58,671</td>
<td>8,157</td>
<td>1,435,645</td>
<td>10,938</td>
<td>6,549 – 9,765</td>
<td>6,034 – 10.9k</td>
</tr>
</tbody>
</table>
The Need For Speed
2,000+ publicly accessible etcd installations yielded 8,781 passwords. @gcollazo details what he found here: elweb.co/the-security-f...

It really is as simple as http://<IP address of etcd instance>:2379/v2(keys/?recursive=true

Here's an example MySQL password found:

```
payment.kubernetes.io/revision":1","spec":  
"pod-template-hash":665149664","spec":  

e":"MySQL_ROOT_PASSWORD","value":"12345"  
"/dev/termination-log","imagePullPolicy":"I
```

10:05 PM - 17 Mar 2018

89 Retweets 140 Likes
2018 - Top Detections - “Scannables”

Apache Struts 2.3.x - CVE-2017-5638, CVE-2017-9791, CVE-2017-9805

Joomla! 3.7.1 - CVE-2017-8917

Jenkins 2.56 - CVE-2017-1000353

MASTER IPCAMERA - CVE-2018-5723 (hardcoded password)

Microsoft SMBv1 - CVE-2017-0143/4/5

Oracle WebLogic 10.3.6, 12.1.x, 12.2.x - CVE-2017-10271

PHP 5.4.2 - CVE-2002-1149, CVE-2012-1823
IPv4 is ... too small

1998 - Bell Labs - Internet Mapping Project
2009 - SHODAN
2011 - Fyodor - Nmap: Scanning the Internet
2011 - Carna botnet “Internet Census of 2012”
2012 - HD Moore - Critical.IO
2012 - University of Michigan (zmap) / CENSYS
2014 - Rob Graham - Masscanning the Internet
Now - ... everybody
CIS #1: Inventory & Control of HW Assets

Hardware Asset Inventory

Active Discovery
Passive Asset Discovery
Use DHCP Logging
Address Unauthorized Assets
Deploy Network Access Control
Utilize Client Certificates
CIS #2: Inventory & Control of SW Assets

Software Asset Inventory

Maintain Inventory of Authorized Software
Ensure Software is Supported
Integrate SW & HW Asset Inventories
Address Unapproved Software
Utilize Application, Library, Script Whitelisting
Segregate High Risk Applications
ITSM & CMDB - Asset Discovery & Mgmt

Extensive discovery capabilities...

internal view... generally require creds

rarely integrated with vulnerability or threat data
Vulnerability Scanners & Asset Discovery

- Provide limited discovery capabilities
  - In practice, network ranges are used
- Scan windows are still a challenge, and may not provide enough information quickly enough
- Depth and completeness favored over quick scans
More Layers... More Complexity

...Yep, we’re making it worse.
IT Asset Management... Security is secondary

HAM: Hardware Asset Management
SAM: Software Asset Management
ITAM: IT Asset Management
ITSM: IT Service Management
Now, Devops.
Visibility ... Fragmented
Visibility is a Major Challenge

Mid Tier - 11 different discovery and inventory tools
Enterprise - 15 different discovery and inventory tools
Average respondent spent about 15 hours a week
More successful respondents spent more (not less) time doing this!
BEST CASE... 60-70% percent of assets covered
Recon as a Process

Asset Discovery - PROCESS utilizing a technique to find new assets
Asset Inventory - COLLECTION of things and their specific attributes
Asset Management - a end to end management PROCESS for assets

(Defender) Recon - PROCESS for preliminary surveying or research of devices, software, or specific vulnerabilities
So you’re saying...

Many RCE vulnerabilities are being scanned
Internet scanning is trivial
Unknown assets are a big problem for larger organizations
Vulnerability scanning helps, but leaves unknown assets
Asset management is foundational but often incomplete

... Recon techniques can help.
Taking on an Adversarial Perspective
What’s different now

Ipv4 Internet Scanning
Databases full of security data
Application everything
Enter... Bug Bounty Recon
Striking Gold!

```
<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /sm/login/loginpagecontentgraber.do HTTP/1.1</td>
<td>HTTP/1.1 200 OK</td>
</tr>
<tr>
<td>Host: svdevems01.direct.gql.yahoo.com</td>
<td>Server: Apache-Coyote/1.1</td>
</tr>
<tr>
<td>User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0</td>
<td>Cache-Control: private</td>
</tr>
<tr>
<td>Accept: text/html,application/xhtml+xml,application/xml;q=0.9,<em>/</em>;q=0.8</td>
<td>Expires: Thu, 01 Jan 1970 00:00:00</td>
</tr>
<tr>
<td>Accept-Language: en-US, en;q=0.5</td>
<td>Set-Cookie: JSESSIONID=86A377A5</td>
</tr>
<tr>
<td>Content-Type: application/x-www-form-urlencoded('X-Ack-Th3gSh3lman-POC',4x4)</td>
<td>X-FRAME-OPTIONS: SAMEORIGIN</td>
</tr>
<tr>
<td>X-Requested-With: XMLHttpRequest</td>
<td>X-Ack-Th3gSh3lman-POC: 16</td>
</tr>
<tr>
<td>Referer: <a href="https://svdevems01.direct.gql.yahoo.com/sm/login.jsp">https://svdevems01.direct.gql.yahoo.com/sm/login.jsp</a></td>
<td>Content-Type: text/html; charset=iso-8859-1</td>
</tr>
<tr>
<td>Cookie: JSESSIONID=D2883399FADA32A1B18E21769FE061A4; B=b86de9c959f3b46d=ubff1appYF38kTvLv9Qsxl8Ba94=;s=qu6i=05LWvR84cFFfGN e42Qc; A0=x1; F=a4vEd0jUyNy5DE1w_LnESbWMLyXkL2dCLiBBp8sGlS.fpwlVptsuA41PMv6Ref7TxMyu v8t11B05xqIIddDGd3s9A--6bUPwz6dGd70Kyk9v0---; uces=f=sl1nct=18743995476hs=2; Y=%2e; YLS=vl1p=16n1;</td>
<td>Content-Length: 7079</td>
</tr>
<tr>
<td></td>
<td>Vary: Accept-Encoding</td>
</tr>
<tr>
<td></td>
<td>Date: Mon, 13 Mar 2017 12:32:42</td>
</tr>
<tr>
<td></td>
<td>Connection: close</td>
</tr>
</tbody>
</table>
```

---

KENNA Security

# RSAC

Matters

RSAC Conference 2018

<table>
<thead>
<tr>
<th>State</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolved (Closed)</td>
<td>High (7 ~ 8.0)</td>
</tr>
</tbody>
</table>

- **Disclosed publicly**: June 17, 2017 8:59am - 0700h
- **Participants**: Udemy
- **Weakness**: Code Injection
- **Bounty**: $300

**SUMMARY BY cha5m**

I discovered a critical information disclosure bug via an exposed Jenkins dashboard located at [https://jenkins101.udemy.com](https://jenkins101.udemy.com). Upon navigating to this address, I was presented with a Github authentication page. After authenticating, I was surprised to find that I had complete read access to the corresponding Jenkins Dashboard. Contained within the dashboard was the complete Udemy source code, including the keys for various Udemy services.

**TIMELINE**

- **cha5m submitted a report to Udemy**: Nov 14th (about 1 year ago)

**Summary:**

I am writing to inform you of a critical information disclosure bug via an exposed Jenkins dashboard located at [https://jenkins101.udemy.com](https://jenkins101.udemy.com). Upon navigating to this address, I was asked to authenticate with my Github account. After authenticating, I was surprised to find that I had complete access to the corresponding Jenkins Dashboard as seen in the screenshot below:

![Screenshot of Jenkins Dashboard](image-url)
Bug Bounties... Finding Targets

WHOIS Data
DNS - Active & Passive
Scanning
Certificates
Application Responses
Bug Bounties & Recon

- Subdomain Bruteforcing & Permutations
- Zone Transfers & NSEC walks
- Querying Historical APIs - WHOIS, DNS
- Scanning Nmap & Masscan (or SHODAN / CENSYS)
- Fingerprinting Services, Applications
The Need for Speed

DONT KNOW WHAT THIS IS

BUT IT LOOKS FAST
Intrigue - Sources (partial list)

aws_ec2_gather_instances
aws_s3_brute
dns_brute_sub
dns_nsecwalk_survey
dns_permute
dns_transfer_zone
email_harvest
masscan_scan
nmap_scan

search_bing
search_censys
search_corpwatch
search_crt
search_github
search_opencorporates
search_shodan
search_sublister
search_whoisology

uri_brute
uri_extract_metadata
uri_gather_ssl_certificate
uri_screenshot
uri_spider
web_account_check
web_stack_fingerprint
whois
whois_org_search
Graph-Based Gathering
<table>
<thead>
<tr>
<th>IP Address</th>
<th>OS/Software</th>
<th>Version</th>
<th>Web Framework</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>44.244.242.173:80</td>
<td>mod_ssl/2.2.21 OpenSSL/1.0.0-fips</td>
<td></td>
<td></td>
<td>Google Analytics; Facebook</td>
</tr>
<tr>
<td>44.244.253:80</td>
<td>Apache/2.2.21 (Unix) DAV/2</td>
<td></td>
<td></td>
<td>Google Analytics; Facebook</td>
</tr>
<tr>
<td>44.244.238:80</td>
<td>Microsoft-IIS/7.0</td>
<td></td>
<td></td>
<td>Jquery; Wordpress; Cloudflare</td>
</tr>
<tr>
<td>44.244.240:80</td>
<td>nginx</td>
<td>PHP/5.6.30</td>
<td></td>
<td>Jquery; Wordpress; Cloudflare</td>
</tr>
<tr>
<td>44.244.247:80</td>
<td>Microsoft-IIS/7.0</td>
<td></td>
<td></td>
<td>Jquery; Wordpress; Cloudflare</td>
</tr>
<tr>
<td>44.244.210:80</td>
<td>nginx</td>
<td></td>
<td></td>
<td>Wordpress</td>
</tr>
<tr>
<td>44.244.210:80</td>
<td>Microsoft-IIS/7.0</td>
<td></td>
<td></td>
<td>Wordpress</td>
</tr>
<tr>
<td>44.244.242.173:80</td>
<td>mod_ssl/2.2.21 OpenSSL/1.0.0-fips</td>
<td></td>
<td></td>
<td>Wordpress</td>
</tr>
<tr>
<td>44.244.242.173:80</td>
<td>mod_ssl/2.2.16 OpenSSL/0.9.8a</td>
<td></td>
<td></td>
<td>Wordpress</td>
</tr>
<tr>
<td>44.244.242.173:80</td>
<td>mod_ssl/2.2.21 OpenSSL/1.0.0-fips</td>
<td></td>
<td></td>
<td>Wordpress</td>
</tr>
<tr>
<td>44.244.242.173:80</td>
<td>mod_ssl/2.2.16 OpenSSL/0.9.8a</td>
<td></td>
<td></td>
<td>Wordpress</td>
</tr>
<tr>
<td>44.244.242.173:80</td>
<td>mod_ssl/2.2.21 OpenSSL/1.0.0-fips</td>
<td></td>
<td></td>
<td>Wordpress</td>
</tr>
<tr>
<td>44.244.242.173:80</td>
<td>mod_ssl/2.2.21 OpenSSL/1.0.0-fips mod_jk/1.2.28</td>
<td></td>
<td></td>
<td>Wordpress</td>
</tr>
<tr>
<td>44.244.242.173:80</td>
<td>mod_ssl/2.2.21 OpenSSL/1.0.0-fips mod_jk/1.2.28 DAV/2</td>
<td></td>
<td></td>
<td>Wordpress</td>
</tr>
<tr>
<td>44.244.242.173:80</td>
<td>mod_ssl/2.2.21 OpenSSL/1.0.0-fips mod_jk/1.2.28 DAV/2</td>
<td></td>
<td></td>
<td>Wordpress</td>
</tr>
<tr>
<td>44.244.242.173:80</td>
<td>mod_ssl/2.2.21 OpenSSL/1.0.0-fips mod_jk/1.2.28 DAV/2</td>
<td></td>
<td></td>
<td>Wordpress</td>
</tr>
</tbody>
</table>
D.C. Court: Accessing Public Information is Not a Computer Crime

BY JAMIE WILLIAMS | APRIL 12, 2018

Good news for anyone who uses the Internet as a source of information: A district court in Washington, D.C. has ruled that using automated tools to access publicly available information on the open web is not a computer crime—even when a website bans automated access in its terms of service. The court ruled that the notoriously vague and outdated Computer Fraud and Abuse Act (CFAA)—a 1986 statute meant to target malicious computer break-ins—does not make it a crime to access information in a manner that the website doesn’t like if you are otherwise entitled to access that same information.
Top Detected PHP versions

- PHP/5.4.16
- PHP/7.0.29
- PHP/5.6.30-1~dotdeb+7.1
- PHP/5.3.10-1ubuntu3.25
- PHP/5.6.31
- PHP/5.4.36
- PHP/5.4.45
- PHP/5.4.45-4+deprecated+don't use+deb.sury.org+precise+1
- PHP/5.6.34
- PHP/5.6.30
- PHP/5.3.8
- PHP/5.3.3
- PHP/5.5.9-1ubuntu4.20
- PHP/5.4.7

- PHP/5.3.2
- PHP/5.3.10-1ubuntu3.25
- PHP/5.3.1
- PHP/5.2.6-1+lenny16
- PHP/5.2.6
- PHP/5.2.17; ASP.NET; Yoast WordPress SEO plugin
- PHP/5.2.17
- PHP/5.2.14; ASP.NET; ARR/3.0; ASP.NET; Wordpress
- PHP/5.2.14
- PHP/5.1.6
“Electricity - Powering Stuff Since 1879”

“WordPress 2.7.1; Wordpres API”

`Apache/1.3.31 (Unix) mod_jk/1.2.5 PHP/5.2.17 FrontPage/5.0.2.2634 mod_fastcgi/2.4.2 mod_throttle/3.1.2 mod_ssl/2.8.18 OpenSSL/0.9.7d`

Cisco Stealthwatch 1.0.1

“That would be telling.”
It’s not just external

No one discovery tactic to rule them all

- Local - Plug into the Network -
- Cloud - APIs Provided
- External - Iterative OSINT

- Bringing it all together requires an integration-first approach
  - Each asset with a small set of required data and a dynamic locator
Integrating into your Vulnerability and Risk Management Program
Operationalizing

- Measuring Success - How quickly can you determine if you’re subject to a particular vulnerability or technique

- An automated external recon capability can provide a safety net, and... You can enlist hackers as part of that safety net via Bug Bounty or Vulnerability Disclosure program

- Recon findings should be integrated into risk scoring. If an attacker can find it quickly, the threat is increased
Takeaways

Defender reconnaissance can augment and enhance vulnerability management program - both by finding assets and identifying likely targets

New data sources are available and operationalizable for defenders, and can assist in both asset and vulnerability management

Organization risk management should factor in assets and vulnerabilities discoverable via recon techniques – automatically higher priority

Do you know what software (and versions!) are exposed and scannable?
Putting it into action

**Next Week** Discuss unknown assets in with your asset and vulnerability management teams.

**Three Months** Perform an external discovery for unknown assets using one of the tools we’ve discussed today.

**Six Months** Integrate recon into your asset and vulnerability management processes. Create escalation processes for new assets with vulnerabilities. Consider a Bug Bounty or Disclosure program to provide a safety net.
Thank you!
Thank you for your time!