OUTSIDE LOOKING IN:

USING SHODAN FOR VULNERABILITY SCANNING

Chris Woods, CISSP
Mount Holyoke College
Library, Information, & Technology Services

VULNERABILITY SCANNING DEFINITION

Assessment of computers, systems, networks, and/or applications for *weaknesses* than can be exploited by unauthorized persons and potentially result in loss of **confidentiality**, **integrity**, and/or **availability**.

REASONS TO DO VULNERABILITY SCANS

- Monitor compliance
- Determine where to focus your resources
- Quantify risk to the organization
- Identify vulnerable components of your network
- Resource planning

COMMON SCANNING TOOLS

- NMap
- Nessus
- Qualys (variety of free tools, limited)
- censys.io
- Shodan
- Others (consult your local search engine)

A BIT ABOUT SHODAN

- Search engine of Internet connected devices
- Created by John Matherly
- Launched in 2009
- https://www.shodan.io

A BIT MORE ABOUT SHODAN

- Free (as in beer) or inexpensive
- Simple to use
- Web U/I
- Command line tool
- DuckDuckGo !bang syntax (!shodan)
- API
- Enterprise accounts available

SHODAN'S ADVANTAGES

- Fast
- Objective
- Free or inexpensive

SPEED

- Scanning takes time
- Sequential scanning can fail
- Long TTL often required

OBJECTIVE

- No inside knowledge
- Uniform scanning technique
- No organizational bias
- No inadvertent whitelisting
- Random, non-incremental scanning

FREE OR INEXPENSIVE

- Free unfiltered searches without account
- Free filtered searches with account
- Reports and API with paid account ($49 one time)
- Enterprise accounts start at $19/mo

DATA RETURNED BY SHODAN

- Banner text
- Operating system
- Services running
- Versions
- Roughly equivalent to curl command:

```
curl -I /
'http://example.com'
```

SPOTTING VULNERABILITIES

- Out of support versions
- Out of date packages
- Open ports (where none are expected)
- Unusual subnets
- Vendor default pages (mostly IoT but also printers, etc)

SAMPLE TEXT BANNER

```
HTTP/1.1 302 Found
Date: Tue, 22 Aug 2017 01:28:22 GMT
Server: Apache/2.2.0
Location: https://i.madethis.up/
Content-Length: 214
Content-Type: text/html; charset=iso-8859-1
```

ANY QUESTIONS/COMMENTS SO FAR?

INTERNAL VS. EXTERNAL SCANS

Internal scans originate from a privileged host or vlan (i.e. inside the firewall).

External scans originate from the Internet (i.e. outside the firewall).

UNAUTHENTICATED VS. AUTHENTICATED SCANS

Unauthenticated - no response to auth requests.

Authenticated - responds with valid credentials.

CONSTRUCTING AN EXTERNAL UNAUTHENTICATED SCAN

- Consider where your assets are
- State the null hypothesis
- Scan to disprove the null hypothesis
- Run scan from an external IP against your net

STATING THE NULL HYPOTHESIS

A null hypothesis is a hypothesis that a researcher tries to disprove (e.g. "There are no webservers running in given subnet").

TRY DISPROVING THE NULL HYPOTHESIS

```
for ((i=0;i<=255;i++));
do curl -I -k -X GET 192.0.2."${i}":80;
done
```

NINE HOURS LATER...

It took an average of 2:06 to complete the curl request for each address in the /24 IP range of the query.

126 sec * 254 IP addresses = 32,004 seconds
32,004 / 60 = 533 minutes
533 / 60 = 9 hours

SAMPLE TEXT BANNER

```
HTTP/1.1 302 Found
Date: Tue, 22 Aug 2017 01:28:22 GMT
Server: Apache/2.2.0
Location: https://i.madethis.up/
Content-Length: 214
Content-Type: text/html; charset=iso-8859-1
```

OR YOU COULD DO THIS...

1. Open a web browser
2. Go to shodan.io
3. Login
4. Type the following in search box:

```
net:192.0.2.0/24
port:80
```

WHAT DID WE LEARN?

- Nearly identical results
- Reasonably fresh, mostly
- Results are downloadable
- Ready for parsing
- Pipe to other apps

NOW WHAT?

- Click on **Download Results**
- Choose your format (CSV, JSON, XML)
- Import to spreadsheet
- Process with Python, Perl, etc
- Open tickets in tracking system

USE THE RESULTS TO PLAN

- Identify hosts with impending EOS/EOL issues
- Find hosts affected by specific CVE

```
net:192.0.2.0/24 vuln:CVE-2014-0160
```

- Assign work to sys ad
- Enter work into tracking/ticketing system

QUESTIONS OR COMMENTS?

CONTACT INFO

- cswoods@mtholyoke.edu
- https://www.linkedin.com/in/chris-woods-08449973
- https://github.com/pythonsysad

CONSTRUCTING QUERIES FROM JSON

Use the hierarchy from the results JSON to construct new queries.

Given this JSON snippet:

```
"http": {"redirects": [], "title": "302 Found", "robots": null }
```

Corresponding Shodan query:

```
http.title:"302 Found"
```

SERVICES ON NON-STANDARD PORTS

Use the minus sign to exclude results. In this case, exclude the standard SMTP port.

```
product:postfix -port:25
```

PARSING JSON WITH COMMAND LINE

Use the command line tool to parse downloaded results.

This command:

```
shodan parse --fields ip_str,hostnames
--separator , ~/shodan-export.json
```

Returns:

```
192.0.2.106,i.madethisup.edu,
192.0.2.111,learning.is.gd,
```

SEARCHING FROM THE COMMAND LINE

This command line search:

```
shodan search --fields ip_str,hostnames "product:openssh -port:22 net:192.0.2.0/24"
```

Returns a result set like this:

```
192.0.2.106,i.madethisup.edu,
192.0.2.111,learning.is.gd,
```

COLLECT DATA IN REAL TIME WITH STREAMS

Use streams to gather data from Shodan crawlers as it is collected. See the API docs for full details.

CREATE A NETWORK ALERT

Create a network alert for the desired IP range:

```
shodan alert create "My Alert" 192.0.2.0/24

Successfully created network alert!
Alert ID: HFI66IBBH0X8Z8VQ
```

LIST ALERTS TO GET ID

Obtain the alert ID:

```
shodan alert list

Alert ID Name IP/ Network
HFI66IBBH0X8Z8VQ My Alert
192.0.2.0/30
```

SET UP THE STREAM

JSON results will write to /var/lib/shodan...

```
shodan stream --alert
HFI66IBBH0X8Z8VQ --datadir
/var/lib/shodan
```

ANOTHER NEAT THING

DuckDuckGo has a !bang for Shodan. If DDG is your default search engine, type the following into your search bar:

```
!shodan net:192.0.2.0/24 port:80
```

YET ANOTHER NEAT THING

There are Shodan browser plugins for Chrome and Firefox.

FURTHER READING

- The Complete Guide to Shodan by John Matherly
- API Docs - https://developer.shodan.io/api

THANKS VERY MUCH.

QUESTIONS?

CONTACT INFO

- cswoods@mtholyoke.edu
- https://www.linkedin.com/in/chris-woods-08449973
- https://github.com/pythonsysad