Affordable Security:

Making the most of free tools and data

Craig Merchant

Senior Security Architect, Oracle

splunk>

Disclaimer

During the course of this presentation, we may make forward looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not, be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Additional Disclaimer

I am here as a member of the Splunk community.
I am not here as a representative of Oracle or the Oracle Cloud.
Due to the sensitive nature of some of our customers, I am only authorized to disclose a limited amount of specific information about our environment.

Personal Introduction

- Craig Merchant, Senior Security Architect, Oracle Cloud
- Over 20 years of experience in security, networking, and systems management
- Joined Oracle through the Responsys acquisition in 2014
- Oracle Cloud was announced on June 7, 2012
  - Offerings include SaaS, PaaS, IaaS
  - Growth has been driven by over a dozen acquisitions
- My role: To Build the Better Mouse Trap

Agenda

- Augmenting Nessus vulnerability data with vFeed project
- Dynamic vulnerability scanning using SNMP and ARP tables
- Integrating the Collective Intelligence Framework into Splunk
- Running SCAP security checklists using Splunk and jOVAL
- Gain total visibility into network flows using Argus and Sysdig
- All content available at:
  - https://github.com/SplunkSec/.conf2015

Vulnerability Data Enrichment

splunk>

The Incomplete Picture

- Vulnerability Management is the foundation of a mature security practice
- A successful remediation strategy depends on knowing:
  - Is the vulnerability a local or remote exploit?
  - Can an unauthenticated user perform the exploit?
  - How easy is it to perform the exploit?
  - Is there a known exploit for the vulnerability in the wild?
  - Which patches or updates will reduce the most amount of risk?
- The data contained in the plugins from most vulnerability scanners cannot answer all of those questions

What is vFeed?

- https://github.com/toolswatch/vFeed
- vFeed integrates a wealth of vulnerability data:
  - CVE – Common Vulnerabilities and Exposures
  - CWE – Common Weakness Enumeration
  - CPE – Common Platform Enumeration
  - OVAL – Open Vulnerability and Assessment Language
  - CAPEC – Common Attack Pattern Enumeration and Classification
  - CVSS – Common Vulnerability Scoring System
  - Multiple exploit databases: Exploit-DB, Metasploit, Saint…
  - Multiple vulnerability databases: OSVDB, Bugtraq, NVD
  - Vendor Security Advisories: MSFT, Red Hat, Cisco…

Filling in the Gaps

- vFeed provides mappings between Nessus and CVE IDs
- Mapping the CVSS metrics to CVE IDs answers:
  - Is the vulnerability a local or remote exploit?
  - Can an unauthenticated user perform the exploit?
  - How easy is it to perform the exploit?
- Mapping CVE IDs to public exploit databases answers:
  - Is there a known exploit for the vulnerability in the wild?
- Mapping CVE IDs to CPE entries answers:
  - Which patches or updates will reduce the most amount of risk?

How to Build It

- Install vFeed and create a cron job for daily updates
- Connect to the vFeed SQLite database using the DB Connect 1 app
- Create a lookup in Splunk for each table in vFeed
- Dump each table, sanitize the data, and augment with URL info
- Download Common Information Model (CIM) app and augment the Vulnerability data model with vFeed fields
- Modify the CIM Vulnerability data model to perform required lookups
- Populate the kvstore with vulnerability data
- Build dashboards driven off data model searches

Bonus Value

- vFeed has mappings for Snort IDs to CVE IDs
- Create a search that will find IDS events and:
  - Lookup the CVE IDs associated with each Snort ID
  - Use mvexpand to create a single event per CVE ID
  - Lookup the CVE ID in the Snort event against any CVEs for the dest/ dest_port/protocol in the Vulnerability kvstore
- Alarm on a match

Dynamic Vulnerability Scanning

splunk>

Scheduled Scanning is Risky

- **Missed assets** – Laptops and tablets may not be connected to the corporate network during the scheduled scanning window
- **Fragile infrastructure** – Running large scans can exhaust resources on switches, routers, and firewalls
- **Huge address spaces** – IPv6 networks can be massive and virtually impossible to scan within a typical scanning window
- **Easy evasion** – Predictable scanning windows create opportunities for attackers to hide evidence of their intrusion

Splunk Powers Dynamic Scanning

- Use scripted inputs or SNMP apps to poll network switches for the IP-to-MAC address relationships
- Track the scanning history for each MAC address in a lookup table
- Use that history to launch scans via the Nessus API
- Index the status of each scan with a scripted input
- Generate a CSV report via the API when scans complete

Map IPs to Physical Address

- For IPv4, SNMP atPhysAddress (.1.3.6.1.2.1.3.1.1.2) to enumerate the ARP cache:
  - snmpwalk –c public –v 2c 10.10.10.10 atPhysAddress
    mib-2.3.1.1.2.76.1.192.168.1.93 "3C 38 A6 D9 AA 4C "
    mib-2.3.1.1.2.76.1.192.168.1.95 "A8 9D 21 DA 86 B3 "
- For both, SNMP ipNetToPhysicalPhysAddress (.1.3.6.1.2.1.4.35.1.4):
  - snmpwalk –c public –v 2c 10.10.10.10 ipNetToPhysicalTable
    - IP-MIB::ipNetToPhysicalPhysAddress.2.ipv4."10.0.0.1" = STRING: 0:26:5a:f4:1a:28
    - IP-MIB::ipNetToPhysicalPhysAddress.2.ipv6."fe:80:00:00:00:00:00:00:ca:60:00:ff:fe:e9:1a:e9“ = STRING: c8:60:0:e9:1a:e9

Build Scripted SNMP Input

- Create a search that will find all network switches, dedup the hostname or IP, and write the hostnames/IPs to a lookup table
- Create a simple “for” loop in a bash script:

```
for switch in `cat ../lookups/switches.csv | sed “1d”`
do
  echo “Switch:  $switch
  snmpwalk –c public –v 2c $switch ipNetToPhysicalPhysAddress
done
```

- Schedule the scripted input to run every 5 minutes

Interact with Nessus 6 API

- Login and save authentication token:
  - curl -k -X POST -H 'Content-Type: application/json' -d '{"username":"nessus","password":"ne55us"}' https://10.10.10.10:8834/session 2>&1 | grep -Po '(?<=\"token\":\")[^\"]+'
- List the configured scans on the scanner:
  - curl -k -H 'X-Cookie: token=f99a30c7d590f07880f27aa913ee705955bcaa7b7d51e041' https://10.10.10.10:8834/scans
- Launch a scan with a list of dynamic IPs:
  - curl -k -X POST -H 'X-Cookie: token=f8f0b0821d0ef193d346a2951dbc9e28314bcf232d40e4e7' -H 'Content-Type: application/json' -d '{"alt_targets": ["10.10.10.1,10.10.10.2,10.10.10.3,10.10.10.4"]}' https://10.10.10.10:8834/scans/6/launch
- Export a scan to CSV format:
  - curl -k -X POST -H 'X-Cookie: token=f8f0b0821d0ef193d346a2951dbc9e28314bcf232d40e4e7' -H 'Content-Type: application/json' -d '{"format": "csv"} ' https://10.10.10.10:8834/scans/6/export

How to Build It

- On each scanner, define the necessary policies and scan configs
- Use the Nessus API to find the ID of configured scans
- Create a CIDR match lookup table to assign Nessus IPs and scan IDs to hosts or subnets:

  dest, nessus_ip, scan_id
  “10.10.10.0/24”,”10.10.10.10”,6
  “10.10.11.0/24”,”10.10.10.10”,5

- Create a lookup to track the last scan time for a MAC address and seed it with data that adds random padding to the last_scan time:

  (sourcetype="nessus" AND signature="12053" OR signature="19506") OR sourcetype=snmp_arp | rex field=_raw "(?i)resolves as (?P<hostname>[^$]+)(?=\.)" | rex mode=sed field=dest_mac "s/\s/:/g"| transaction maxspan=1h dest | eval padding=random()/2147483648*86400 | eval last_scan=now()+padding | table last_scan,hostname,dest,dest_mac | outputlookup nessus_last_scan_lookup

How to Build It

- Create a lookup table to queue IPs that need to be scanned
- Create a search that will check incoming ARP events against the last scan lookup, lookup the scanner IP and scan ID, and save those events to the queue
- Create a scripted input to iterate through the Nessus scanners, list the status of scan jobs and index the results
- Create a lookup table to track completed scan jobs and export status
- Create a script that can iterate through the scan job lookup table and use the Nessus API to export the scan in CSV format
- Create a file monitor input on the Nessus scanner that will index all CSV files found in each user's "files" directory

Integrating Threat Intelligence

splunk>

Why Threat Intel?

- Today's attackers are incredibly sophisticated and agile
- Threat Intel providers take the offensive against hackers
- Data is updated more frequently than traditional AV vendors
- Indicators of Compromise (IOCs) can be found in a wide range of security data types

What is CIF?

- https://code.google.com/p/collective-intelligence-framework/
- CIF collects, normalizes, qualifies, and categorizes free and commercial threat intelligence feeds
- Supported data types:
  - File Hashes (MD5s)
  - IPv4 Addresses
  - Host Names
  - Domain Names
  - URLs (MD5s)
- v1 stores each category of threat in a separate Postgres database table

What Data can it Enrich?

- Network Intrusion Detection events (IPv4, Host/Domain names)
- Network Flow events (IPv4, Host/Domain Names, URL)
- Proxy Logs (IPv4 Host/Domain names, URL)
- Firewall Logs (IPv4, URL)
- Endpoint Protection Solutions (File Hash, IPv4, Host/Domain names)
- Anti-Malware solutions (File Hash, IPv4, Host/Domain names, URL)

How to Build It

- Install CIF v1 and configure threat intel feeds
- Connect to the Postgres database using DB Connect App
- Create Splunk lookup tables for each classification type – botnet, phishing, spam, etc.
- Create searches to periodically dump the database tables into lookup tables
- Create automatic lookups to enhance log data
- Optional: Add a menu context item to check fields against the HTTP API: http://eyeis.net/2012/04/querying-cif-data-from-splunk/

Automated Security Configuration Monitoring with SCAP

splunk>

Value and Challenges

- Security configuration monitoring can measure the compliance of systems and applications against a set of enterprise security standards
- The human “checklist” approach is slow, expensive, and prone to error
- Creating customized scripts to perform configuration checks is expensive to develop and maintain and difficult to instrument and report on

What is SCAP?

- SCAP: Security Content Automation Protocol
  - XCCDF: Extensible Configuration Checklist Definition Format
  - OVAL: Open Vulnerability and Assessment Language
- Public repositories of SCAP content:
  - https://web.nvd.nist.gov/view/ncp/repository
  - http://usgcb.nist.gov/
  - http://iase.disa.mil/stigs/scap/Pages/index.aspx
  - https://benchmarks.cisecurity.org/downloads/
- SCAP scanners:
  - Free: http://www.open-scap.org/
  - Commercial: http://jovalcm.com/

What Does the Data Look Like?

- XCCDF scan using OpenSCAP:

  Title Uninstall squid Package
  Rule uninstall_squid
  Ident CCE-26977-9
  Result pass

  Title Disable snmpd Service
  Rule disable_snmpd
  Ident CCE-26906-8
  Result pass

- OVAL scan using OpenSCAP:

  Definition oval:ssg:def:959: false
  Definition oval:ssg:def:957: true
  Definition oval:ssg:def:955: true

How to Build It

- jOVAL – Install on central scanning host
- OpenSCAP – yum install openscap openscap-utils openscap-content
- Index the XCCDF or OVAL XML files
- Extract important fields from the rule objects – id, title, description, severity, weight, and fixtext
- Create a lookup table or kvstore containing the policy name and the rule objects above
- Create automatic lookup to enrich scanner logs with rule data
- Create an app for the policy files and scripted inputs
  !!! You must change the file extension from XML or you will break Splunk !!!

Improving Visibility With Network Flows

splunk>

Getting Flow Security Right is Hard

- What collection method to use – Netflow protocols or taps/SPAN ports?
- Where to collect the data – Internet connections, internal switches, or individual hosts?
- What flow metrics need to be collected for security or ops?
- What flow aggregation and filtering method to use?
- What events are worth sending to Splunk?
- How can flows be used to complement other security data?

What is Argus?

- http://qosient.com/argus/
- Free, open source tool capable of generating flow records from both flow protocols (Netflow, IPFIX) and raw network streams
- Capable of capturing all, some, or none of the packet payload
- Supports a rich set of flow metrics
- Allows tagging of flows based on CIDR match
- Key architecture components:
  - argusd: Argus flow collector daemon
  - radium: Collects and dedups flows from multiple Argus daemons
  - Argus clients: Individual client applications to search and manage Argus data

What Can Argus Do?

- **ralabel**: Uses CIDR network and host matches to apply labels to flows
- **radark**: Looks for high numbers of ICMP unreachable messages to detect scanning behavior
- **rapolicy**: Monitors network traffic for violations of Cisco ACL rules
- Using advanced jitter analysis, Argus can detect keystrokes in encrypted tunnels – excellent for detecting back doors
- Regex pattern matching on payload can identify SSH on unusual ports

How to Build It

- Compile argusd for collection servers
- Compile argus-clients
- Convert Splunk IPv4 threat intel lookup table to Argus label file format
- Configure Argus client searches
- Create Splunk searches to index output of Argus clients
- Advanced: Use radium to save X minutes/hours/days of all flow records and selectively index the data in Splunk

But Wait… Who Dunnit?

- Argus has no user or process attribution capabilities – all its sees is the network
- Enter Sysdig – “Think of sysdig as strace + tcpdump + htop + iftop + lsof + awesome sauce”
  - http://www.sysdig.org
- Sysdig is a Linux kernel module that can gather data about:
  - User activity such as shell commands and network connections
  - Process activity
  - System performance – I/O, CPU, network, and memory
  - Limited Windows/OS X support

How to Build It

- Compile and install sysdig
- Create a scripted input to capture user, UID, PID, PPID, process names and arguments, and IP/port/protocol data for each network connection:
  - sysdig evt.type=connect and fd.lip!=127.0.0.1 –p"%evt.rawtime.s,%evt.type,%fd.cip,%fd.cport,%fd.sip,%fd.sport, %fd.l4proto,%proc.name,%proc.args,%proc.pid,%proc.ppid,%proc.pname,%user.uid,%user.name"
- Create searches for flow/IDS/firewall and sysdig events and bind them together with the "transaction" command
- Build alarms and reports that take advantage of the attribution metadata

Final Thoughts

splunk>

Making the Most of Free

- All of the ideas presented require basic Splunking skill – lookups, transactions, data models, and database connections
- The vFeed and CIF projects provide excellent metadata enhancement capabilities without consuming your Splunk license
- Dynamic vulnerability scanning and SCAP scanning can dramatically improve vulnerability management with a trivial license cost
- Argus offers a number of security client tools that offer great visibility with minimal license consumption
- Sysdig allows security to link network events with the users and programs responsible for them

Future Investigations

- Project Artillery:
  - https://www.trustedsec.com/artillery/
  - A python-based low-interaction honeypot that can be deployed and managed with Splunk
- Google Rapid Response (GRR)
  - https://github.com/google/grr
  - A comprehensive incident response framework that could be automated using data stored in Splunk, such as Notable Events in the Enterprise Security app

Get It at Github

- https://github.com/SplunkSec/.conf2015

Questions?

THANK YOU

splunk>