Detecting web fraud

Applicability

- Product: Splunk Cloud Platform or Splunk Enterprise
- Feature: Search

Scenario

You are an analyst responsible for your organization’s overall security posture. You need to be able to monitor your environment for activity consistent with common attack techniques bad actors use when attempting to compromise web servers or other web-related assets.

When developing a strategy for preventing fraud in your environment, it’s important to look across all of your web services for evidence that attackers are abusing resources to enumerate systems, harvest data for secondary fraudulent activity, or abuse terms of service.

These searches look for evidence of common internet attack techniques that could be indicative of web fraud in your environment, for example account harvesting, anomalous user clickspeed, and password sharing across accounts.

Detection searches

► Account harvesting

To run this search, you'll need to ingest a dataset that provides visibility into the email address used for the account creation. Common data sources used for this detection are customized Apache logs or Splunk Stream.

Fraudsters often create many user accounts on a website before carrying out a campaign. This example search shows how to detect a many-account creation hosted on a Magento2 e-commerce platform, where the fraudster is using email addresses from a single email domain. You'll need to adjust the platform details to fit your environment.

This example narrows the search down for performance reasons, focusing on the single web page that hosts the Magento2 e-commerce platform (via URI) used for account creation, the single http content-type to grab only the user's clicks, and the http field that provides the username, form_data. The search looks for username and email domain, then for numerous account creations per email domain.

This search has been loosely written to help attribute risk or synthesize relevant context by detecting anomalous behavior. To improve the fidelity of this search you can include specifics within your environment, such as a device ID that may be present in your dataset. Other modifications you could make include considering whether the large...
number of registrations are occurring from a first-time seen domain, extending the search window to look further back in time, or calculating the average per hour/day for each email domain to look for an anomalous spikes. You can also use Shannon entropy or Levenshtein Distance to consider the randomness or similarity of the email name or email domain, as the names are often machine-generated.

```
| search (http_content_type=text* sourcetype=stream:http uri="/magento2/customer/account/loginPost")
| rex field=form_data "login\\[username\\]=\(?<Username>[^&|^$]+\)"
| search Username=*
| rex field=Username "@\(?<email_domain>.*)"
| stats dc(Username) AS UniqueUsernames list(Username) AS src_user BY email_domain
| where (UniqueUsernames > 25)
```

► **Anomalous user clickspeed**

To run this search, you'll need to ingest a dataset that includes clickstream data for each user click on the website. The data must have a time stamp and must contain a reference to the session identifier being used by the website. Common data sources used for this detection are customized Apache logs, customized Microsoft IIS logs, or Splunk Stream.

This search examines web sessions to identify those where the clicks occur too quickly for a human or occur with a near-perfect cadence (high periodicity or low standard deviation). This activity is suspicious because it resembles a script-driven session.

The time stamp data, together with a reference to the session identifier being used by the website, ties the clicks together into clickstreams. This value is usually found in the http cookie. With some tuning, a version of this search could be used in high-volume scenarios, for example scraping, crawling, application DDOS, credit-card testing, or account takeover.

False positives from this search may occur since this search has been loosely written to help attribute risk or synthesize relevant context by detecting anomalous behavior. You can improve its fidelity by including specifics within your environment.

```
| search (http_content_type=text* sourcetype=stream:http)
| rex field=cookie "form_key=(?<session_id>\w+)"
| streamstats window=2 current=1 range(_time) AS TimeDelta BY session_id
| where (TimeDelta > 0)
| stats count stdev(TimeDelta) AS ClickSpeedStdDev avg(TimeDelta) AS ClickSpeedAvg BY session_id
| where ((count > 5) AND ((ClickSpeedAvg < 0.5) OR (ClickSpeedStdDev < 0.5)))
```

► **Password sharing across accounts**

To run this search, you'll need to ingest a dataset that includes username and password information for your websites. Tokenized or hashed passwords can be used and are preferable to clear-text passwords. Common data sources used for this detection are customized Apache logs, customized Microsoft IIS logs, or Splunk Stream.

A common password across user accounts generally indicates that the users are choosing poor passwords or that a fraudster has a common password across multiple accounts embedded within a script. This example shows how
to identify user accounts that share a common password that users submit to the website hosting the Magento2 e-commerce platform (commonly found in the HTTP form_data field). You'll need to adjust the platform details to fit your environment.

False positives from this search may occur since this search has been loosely written to help attribute risk or synthesize relevant context by detecting anomalous behavior. You can improve its fidelity by including specifics within your environment.

```
| search (http_content_type=text* sourcetype=stream:http uri=/magento2/customer/account/loginPost*)
| rex field=form_data "login\[username\]=(?<Username>[^&|^$]+)"
| rex field=form_data "login\[password\]=(?<Password>[^&|^$]+)"
| stats dc(Username) AS UniqueUsernames values(Username) AS user list(src_ip) AS src_ip BY Password
| where (UniqueUsernames > 5)
```

---

### Investigative searches

► **Get emails from specific senders**

To run this search, you'll need to ingest ingest email logs or capture unencrypted email communications within network traffic, and populate the Email data model. Content developed by the Splunk Security Research team requires the use of consistent, normalized data provided by the Common Information Model (CIM). For information on installing and using the CIM, see the Common Information Model documentation.

This search returns all the emails from a specific sender. You'll need to include the sender's details in the placeholder shown.

```
| from datamodel Email.All_Email
| search src_user=<src_user>
```

---

### Additional resources

This use case is included within Splunk Enterprise Security, a Splunk app that provides prebuilt content and searches to help answer root-cause questions in real-time about malicious and anomalous events in your IT infrastructure. In addition, Splunk Enterprise Security provides a number of other searches to help you detect abuse attempts within your environment, including:

- Detecting host redirection attacks
- Detecting Netsh attacks
- Detecting SNICat SNI data exfiltration