Nidhogg

Controlling

Ives Schneider

Index

1. Info

Dieses Dokument beschreibt das Controlling welches währedn der Arbeit eingesetzt wurde.
Es beinhaltet das Zeitmanagement sowie die Testcases für das entwickelte Programm.

2. Zeitplan

3. Meilensteine

KW36 | Projektantrag

Der erste Meilenstein konnte ohne Probleme eingehalten werden.
Nach einigen Verbesserungen wurde der Projektantrag am 13.09.2019 angenommen.

KW38 | Evaluation

Die Produktevaluation konnte bereits etwas früher abgeschlossen werden.
Dies konnte aufgrund bereits vorhandenem Wissen der zu evaluierenden Software geschehen.

KW45 | Issues

Die Issues (Aufbau des Programms), nahm mehr Zeit in Anspruch als anfangs geplant.
Allerdings ist dies nicht weiter schlimm, da beim gekenzeichneten Meilensteins, bereits ein POC des Tools vorhanden war.

KW44 | Einbindung

Der Meilenstein verschob sich ebenfalls um ein paar Tage, dies war aufgrund einer nicht Komplett fertigen Software Konfiguration, welche danach fertiggestellt wurde.

KW47 | Abschluss

Obwohl einige der anderen Meilensteine sich etwas verschoben hatten, konnte der Abschluss Termingerecht eingehalten werden.

4. Testing

4.1. Case #1

Title	Start & Listen
Beschreibung	Nidhogg startet und hört auf den gewúnschten Port (8080)

Config	`webserver:`
	`  ip: "0.0.0.0"`
	`  port: 8080`

Eingabe	`./nidhogg`
Check	`curl localhost:8080`
Soll	Request antwortet mit 200
Ist	` HTTP/1.1 200 OK content-length: 1994 content-type: text/html date: wed, 13 Nov 2019 17:20:53 GMT `
Erfolgreich	Ja

4.2. Case #2

Title	Portscan
Beschreibung	Der eigene Computer wird als zu überwachende Maschine eingetragen. Danach wird ein Port geöffnet, welcher als geschlossen angegeben ist. Hierbei wird die automatische Benachrichtigung getestet. d.H. der Portscan sollte nach angegebener Zeit automatisch durchgeführt werden.

Config	`portspecs:`
	`  - name: yorha`
	`  - id: 8081`
	`  state: closed`

Eingabe	n/a
Check	Automatische Mail wird abgewartet.

Soll	`Time: [Nov 17 17:15:49]`
	
	`[Scan]`
	`Pass: 0`
	`Fail: 1`
	`Analysis:`
	`IP: 10.0.0.25`
	`Result:`
	`[ALERT] Port: 8081 - State: OpenButClosed`

Ist	`Time: [Nov 17 17:15:49]`
	
	`[Scan]`
	`Pass: 0`
	`Fail: 1`
	`Analysis:`
	`IP: 10.0.0.25`
	`Result:`
	`[ALERT] Port: 8081 - State: OpenButClosed`

Erfolgreich	Ja

4.3. Case #3

Title	Arpscan
Beschreibung	Es wird während dem Betrieb eine zusätzliche VM im Netzwerk aufgeschalten. Nach dem hochfahren, sollte die IP einen Broadcast ARP-Request durchführen um den DHCP Server zu finden.

Config	`arpscan:`
	`  enable: trye`
	`  interface: "eth0"`
	`  db: "/etc/nidhogg/arp.db"`
	`  timeout: 50`
	`  mac:`
	`  - 00:00:00:00:00`

Eingabe	n/a
Check	Automatische Mail wird abgewartet.
Soll	`[Time] New device found: 00:50:56:93:58:5c`
Ist	`[[Nov 17 17:20:01]] New device found: 00:50:56:93:58:5c`
Erfolgreich	Ja

4.4. Case #4

Title	Web - Authentication
Beschreibung	Das Webinterface muss eine Authentifizierung besitzen, welche nach einem erfolgreichem Login den Zugang zu weiteren Funktionalitäten bietet. = Hier wird getestet, ob die Informationen welche in config.yml angegeben sind, respektiert werden.

Config	`webserver:`
	`  enable: true`
	`  username: "admin"`
	`  password: "hunter2"`
	`  address: "0.0.0.0"`
	`  port: "8080"`

Eingabe	`Username: admin`
	`Password: admin`
	`-`
	`Username: admin123`
	`Username: blah`
	`-`
	`Username: 1=1 or 1;--`
	`Password: 1=1 or 1;--`

Check	Manueller check
Soll	Login sollte nicht erfolgreich sein
Ist	Es wird kein Session-Token erstellt.
Erfolgreich	Ja

4.5. Case #5

Title	Deaktivierung von Features
Beschreibung	In der config.yml wird der Arp-Scan deaktiviert. Der Arpscan sowie die Url /arp sollte nun nicht mehr verfügbar sein.

Config	`arpscan:`
	`  enable: false`
	`  interface: "wlp58s0"`
	`  db: "/etc/nidhogg/arp.db"`
	`  timeout: 500`
	`  mac:`
	`  - "00:00:00:00:00:00"`

Eingabe	n/a
Check	Es wird eine zusätzliche Maschine hochgefahren. Manueller check auf /arp und timeout abwarten.
Soll	Es sollte kein Alert-Mail versendet werden.
Ist	Es wird kein Arp-alert versendet.
Erfolgreich	Ja

4.6. Case #6

Title	Portscan: Unmögliche Konfiguration
Beschreibung	In der Portspec wird angegeben, dass ein Port sowohl offen wie auch geschlossen sein muss.
Config	

```
portspecs:
    - name: artoria
      ports:
        - id: 22
          state: open
        - id: 22
          state: closed
```

Eingabe	./nidhogg
Check	n/a
Soll	Es sollte die erste Value genommen werden.
Ist	Nidhogg meldet einen Fehler sobald SSHd nicht mehr aktiv ist.
Erfolgreich	Ja

4.7. Case #7

Title	Portscan: Unbekannter Port wird geöffnet
Beschreibung	Ein zusätzlicher Pora (30718)t wird mit netcat geöffnet und simmuliert eine binding shell.

Config	`portspecs:`
	`  - name: artoria`
	`  ports:`
	`  - id: 22`
	`  state: open`

Eingabe	./nidhogg
Check	Das Webinterview /port wird manuell aufgerufen & die automatische Mail wird abgewartet.
Soll	nidhogg sollte den unbekannte Port melden.
Ist	10.0.0.36 - Port: 30718 - State: Open --- Time: [Nov 17 16:58:41] Pass: 0 Fail: 1 Analysis: IP: 10.0.0.36 Result: [ALERT] Port: 30718 - State: ClosedButOpen
Erfolgreich	Ja

4.8. Case #8

Title	PRTG
Beschreibung	PRTG wird eine Meldungen senden, dass sich ein port geändert hat. Dadurch wird ein Mail aktiviert, welches die letzte Nachricht von dem Host in Splunk einbetten sollte.
Config	Es wird ein notification Trigger auf nginx gelegt. Sobald der State sich ändern sollte, wird wird die URL: nidhogg.hosts.i-401.xyz mit den Sensor informationen via GET aufgerufen.
Eingabe	`systemctl stop nginx`
Check	Nginx wird manuell gestoppt.
Soll	nidhogg sollte ein Mail mit Portscan inhalt senden.
Ist	Host: nginx changed sensor: HTTP to state: UP Please investigate! Pass: 0 Fail: 1 Analysis: IP: 10.0.0.201 Result: [ALERT] Port: 22 - State: ClosedButOpen
Erfolgreich	Ja

4.9. Case #9

Title	Splunk
Beschreibung	Der Output der letzten Splunk info wird via Mail gesendet, sobald PRTG einen Host spezifischen Sensor meldet.
Config	Es wird ein notification Trigger auf nginx gelegt. Sobald der State von Load sich ändern sollte, wird wird die URL: nidhogg.hosts.i-401.xyz mit den Sensor informationen via GET aufgerufen.
Eingabe	`curl nidhogg.hosts.i-401.xyz:8080/sensor/nginx/load/up`
Check	n/a
Soll	nidhogg sollte ein Mail mit Splunk content senden.
Ist	Host: nginx changed sensor: Load to state: up Please investigate! Last Splunk messages: [2019-11-17 17:23:50.000 UTC] Nov 17 17:23:50 nginx sudo: pam_unix(sudo:session): session closed for user root
Erfolgreich	Ja