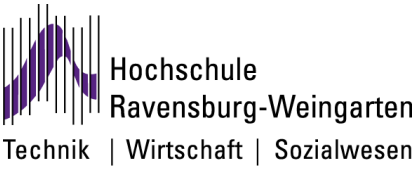


Dokumentation

zur Vorlesung Systemadministration
im Bachelor Studiengang Angewandte Informatik

Wintersemester 2016 / 2017
bei Herrn Prof. Dr. rer. nat. Eggendorfer

Umsetzung von einem Honeypot

Michael Stroh
Matrikelnr. 24972

Daniel Schwenk
Matrikelnr. 24961

24. Oktober 2016

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Motivation 1
1.2 Ziele 2
1.3 Eigene Leistung 2

2 Anforderungen **3**
2.1 Muss-Kriterien 3
2.2 Soll-Kriterien 4
2.3 Kann-Kriterien 4

3 Marktanalyse **5**
3.1 HoneyDrive 5
3.2 Kippo 5
3.3 Honeyd 6
3.4 Glastopf 6

4 Lösungansätze **7**

Literaturverzeichnis **8**

A Anhang **9**

1 Einleitung

Das Internet und die Digitalisierung, die in alle Lebensbereiche Einzug hält, verändern Gesellschaft, Wirtschaft und Kultur. Egal ob im privaten oder beruflichen Umfeld, ständig sind wir von Computern in Form von Arbeitsgeräten, Smartphones oder anderen Geräten umgeben. Diese Verbreitung sowie die Vernetzung von Geräten untereinander wird in den nächsten Jahren im Zuge der „Internet-der-Dinge-Evolution“ weiter drastisch zunehmen.

Ein oft vernachlässigter Aspekt ist hierbei das Thema „IT-Sicherheit“. Keine Software ist frei von Fehlern und Sicherheitslücken. Falsch konfigurierte Dienste und Software, die nicht regelmäßig aktualisiert wird, sind ein leichtes Ziel für Angreifer. Durch die zunehmende Vernetzung wird das Thema IT-Sicherheit in Zukunft weiter an Bedeutung gewinnen.

Um eine Infrastruktur, egal ob im privaten oder geschäftlichen Bereich, vor möglichen Angriffen zu schützen bedarf es eines immer größeren Aufwandes.

1.1 Motivation

Die Gewährleistung der IT-Sicherheit ist mittlerweile eine immens wichtige, wenn nicht sogar die wichtigste Anforderung an eine intakte IT-Infrastruktur. Entsprechend sollte ein Systemadministrator über ein breites Spektrum an Wissen im Bereich der IT-Sicherheit besitzen sowie in der Lage sein, mögliche Angriffsszenarien frühzeitig zu erkennen.

Das Konzept eines Honeypots, also einen potentiellen Angreifer nicht nur vor eigentlich wichtigen System fernzuhalten, sondern auch noch von seinem Wissen zu profitieren, stellt dabei einen hochspannenden Ansatz dar. Dieser Ansatz soll dem Projektteam helfen, Wissen über mögliche Angriffsszenarien und Vorgehensweisen zu erlangen, um so die Sicherheit von bestehenden und zukünftigen Infrastrukturen gewährleisten zu können.

1.2 Ziele

Ziel dieser Arbeit ist es, ein System zu entwickeln, das als Honeypot dient. Dieser Honeypot soll eingesetzt werden, um Einblicke in die Vorgehensweise eines Angreifers zu bekommen. Das System stellt dazu ein vermeintlich leicht angreifbaren Webserver sowie SSH-Server dar.

Jegliche Zugriffe und Aktivitäten die ein Angriff hinterlässt werden protokolliert und ausgewertet. Das hierbei gewonnene Wissen soll in Form von IT-Sicherheitsmaßnahmen in bestehende und künftige IT-Infrastrukturen einfließen.

- Primärziel: einsatzfähige(r) Honeypot(s) - sicher, authentisch und lehrreich

1.3 Eigene Leistung

Der Hauptbestandteil dieses Projekts liegt in der Inbetriebnahme und Bereitstellung eines Honeypots, sowie die Integration desselben in eine für einen potentiellen Angreifer authentisch erscheinenden Umgebung. Dabei liegt das Hauptaugenmerk darauf, die Sicherheit des Systems zu gewährleisten. Die Dokumentation der Infrastruktur, der stattgefundener Angriffe sowie deren Auswertung stellen einen wichtigen Bestandteil dar. Dieses Wissen dient dem Projektteam zukünftig zur Absicherung von IT-Infrastruktur.

- Bereitstellung einer authentischen Umgebung für den Honeypot
- Gewährleistung der Sicherheit für das eigene und das globale Netz
- Inbetriebnahme des Honeypots selbst
- automatisierte Auswertung von Log-Files durch Skripte
- aus Auswertung erfolgt Erarbeitung und Ausweitung von Sicherheitsrichtlinien (Passwortrichtlinien, Firewallregeln)
- Dokumentation von Honeypot inklusive Umgebung, Angriffsphase und Ergebnisse

2 Anforderungen

Die Anforderungen an das Honeypot-System werden in Muss-, Kann- und Soll-Kriterien unterteilt.

2.1 Muss-Kriterien

- Honeypot ist über das Internet erreichbar
- Honeypot darf keinerlei Gefahr für andere Systeme darstellen
- Honeypot muss jederzeit deaktivierbar sein
- Angreifer darf keinerlei Möglichkeit zur Interaktion mit dem Host-Betriebssystem haben
- Angreifer darf keine bzw. nur gefälschte Antworten auf Anfragen erhalten
- Honeypot muss mindestens einen, besser jedoch mehrere Dienste, wie beispielsweise HTTP, SSH oder FTP, simulieren/anbieten
- Honeypot muss ein realistisch wirkendes Angriffsziel darstellen
- Angriffe werden geloggt
- ein- und ausgehender Netzwerkverkehr muss (überwacht und) geloggt werden
- protokollierte Daten dürfen durch Angreifer nicht verändert werden können

2.2 Soll-Kriterien

- automatische Benachrichtigung, wenn System angegriffen wird
- Protokollierung und ggf. Forwarding von Log-Files dürfen für den Angreifer nicht sichtbar sein
- automatisierte Auswertung von Logdaten
- Logdateien / ausgewertete Daten werden automatisch separat gespeichert (extra System, Cloud-Speicher)

2.3 Kann-Kriterien

- geringer Stromverbrauch von Honeypot-System
- kostengünstiger Versuchsaufbau
- Reverse DNS-Lookup von Angreifer-IP-Adresse(n)
- Simulation weiterer Geräte (Router, Firewall, PC)
- Honeypot simuliert offenes WLAN-Netz / Fake-Access-Point

3 Marktanalyse

Eine Marktanalyse zeigt, dass eine Vielzahl an verschiedenen Honeypot-Paketen, Skripten und Konfigurationen mit sehr unterschiedlichen Eigenschaften verfügbar sind. Darunter befinden sich sowohl kommerzielle als auch freie Lösungen.

3.1 HoneyDrive

Mit HoneyDrive existiert eine Honeypot-Linux-Distribution auf Basis von Xubuntu Desktop 12.04.04 LTS. Diese Linux-Distribution bringt 10 vorinstallierte und vorkonfigurierte Honeypot-Pakete wie Kippo SSH Honeypot, Glastopf Web Honeypot oder Amun Malware Honeypot mit. Eine ausführliche Auflistung ist unter [1] gegeben. Die Distribution wird als OVA-Datei angeboten und kann so unter einer Virtualisierungssoftware ausgeführt werden. Neben den vorinstallierten Honeypot-Paketen sind des weiteren unter anderem ein Web- und Datenbankserver sowie wie PHPMyAdmin vorinstalliert. Diese Linux-Distribution ermöglicht so einfach und schnell ein Honeypot-System aufzusetzen.

Das große Manko ist hier der veraltete Software-Stand. Die letzte Aktualisierung fand im Jahre 2014 statt. Dies, sowie der Overhead an vorinstallierten und vorkonfigurierten Diensten, birgt die Gefahr, dass ein potenzieller Angreifer über eine Lücke das Hostsystem kompromittieren oder übernehmen kann.

3.2 Kippo

Kippo ist ein SSH-Honeypot, entworfen um Bruteforce-Attacken sowie die komplette Interaktion des Angreifers mit der Shell zu protokollieren. Konnte sich ein Angreifer durch Eingabe der vorbestimmten Kombination aus Benutzer und Passwort einloggen, wird ihm von Kippo ein virtuelles System offengelegt. In diesem System kann der Angreifer wie gewohnt agieren [2].
Um der Anforderung der automatischen Benachrichtigung des Projektteams bei einem Angriff gerecht zu werden, gilt es Logdateien automatisiert zu analysieren und auszuwerten. Wird ein Bruteforce-Angriff erkannt, wird das Projektteam via Email benachrichtigt. Diese Anforderung kann Kippo ohne Anpassungen nicht leisten.

Ein wesentlicher Nachteil von Kippo ist die Tatsache, dass sich Tools zu seiner Erkennung im Umlauf befinden. Entsprechend versierte Angreifer werden Kippo daher frühzeitig erkennen.

3.3 Honeyd

Honeyd wird von unix-artigen Betriebssystemen unterstützt. Er ist ein Daemon, der virtuelle Hosts in einem Netzwerk erzeugt. Diese Hosts können das Vorhandensein spezieller Betriebssysteme und Services simulieren, indem sie mit authentischen Antwortpaketen auf etwaige Anfragen, insbesondere Fingerprint-Pakete reagieren. Honeyd eignet sich besonders für die Ablenkung eines Angreifers und die Verschleierung der wirklichen Infrastruktur. Ebenso dient er als Warnsystem, da jeder Zugriffsversuch auf einem der durch Honeyd erzeugten Hosts ein Hinweis auf unerwünschte Aktivitäten innerhalb der Infrastruktur darstellt [3].

Honeyd eignet sich nicht ohne Weiteres für die Aufzeichnung komplexerer Angriffe, insbesondere solcher, die auf dem System selbst stattfinden, bietet jedoch die Möglichkeit weitere Geräte zu simulieren und somit zur Authentizität der Infrastruktur des Projektteams beizutragen.

3.4 Glastopf

Glastopf ist ein als Webserver getarnter Honeypot. Dieses System nutzt den Umstand, dass viele Angreifer unter Zuhilfenahme von Suchmaschinen nach Schwachstellen auf Webservern suchen, indem es sich selbst bei den gängigsten Vertretern registriert. Dabei wird Fläche für gängige, webbasierte Angriffe wie SQL-Injections, Remote-Code-Execution, File-Inclusion et cetera, geboten. Von einem Angreifer eingeschleuster Code, wird in einer Sandbox ausgeführt. Alle Verbindungen und Angriffsversuche werden geloggt und in einer Datenbank protokolliert [4].

Durch die Mithilfe von Webcrawlern ist es mit Glastopf möglich für die eigenen Schwachstellen Reklame zu betreiben und somit in kürzerer Zeit eine größere Menge an Angriffen auf das System zu lenken. Die optische Aufmachung der Glastopf-Startseite trägt allerdings dazu bei, dass Angriffe in sehr hohem Anteil nur automatisiert und kaum oder gar nicht in individuell gezielter Form stattfinden werden.

4 Lösungansätze

In den Anforderungen wurde definiert, dass das Honeypotsystem einen, oder besser mehrere Dienste anbieten soll. Um ein realistisches Angriffsziel abzugeben und die Dienste im Internet bereit zu stellen, wird ein Hostsystem mit öffentlicher IP-Adresse auf der Infrastruktur des Rechenzentrums aufgesetzt.

Aufgrund der begrenzten Ressourcen ist die Eigenentwicklung von Honeypotdiensten nicht realisierbar. Ein Lösungsansatz für die Bereitstellung von einem SSH-Honeypot ist der Einsatz von Kippo. Damit wird ein SSH-Dienst, der nach erfolgreichem Login ein virtuelles System simuliert, bereitgestellt. Die von Kippo erzeugten Logfiles gilt es automatisiert auszuwerten. Dies wird mit Hilfe von einem Bash-Skript bewerkstelligt, dass IP-Adressen, Benutzernamen und Passwörter extrahiert. Aus den extrahierten IP-Adressen werden zyklisch Firewallregeln für iptables erzeugt, um zukünftige Angriffe von dieser IP zu blockieren.

Zur Bereitstellung von Diensten wie FTP oder HTTP wird ein separates Honeypot-Werkzeug eingesetzt, dass über eine Evaluation einer Anzahl an in Frage kommenden Systeme ausgewählt wird. Um Logfiles und ausgewertete Daten zu archivieren, werden diese automatisch komprimiert und auf einem Cloud-Speicher wie GoogleDrive abgelegt. Eine Auswertung, aber auch Aufbereitung von Systemweiten Logfiles erfolgt mit Werkzeugen wie Logwatch oder Graylog. Eine Benachrichtigung im Falle eines Zugriffs auf die Honeypotdienste kann mit dem Werkzeug “inotifywait” umgesetzt werden. Damit lassen sich Logfiles oder anderen Daten und Verzeichnisse auf Änderungen prüfen, um daraufhin eine Aktion wie den Versand einer Benachrichtigungsemail anzustoßen.

Um das System bestmöglich abzusichern werden nicht benötigte Dienste deaktiviert. Der Zugriff auf das Hostsystem für das Projektteam erfolgt via Public-Key-Authentifizierung über SSH. Dieser SSH-Dienst ist unabhängig vom Honeypot-SSH-Dienst.

Literaturverzeichnis

[1] HoneyDrive: *Honeypots in a box!* http://bruteforce.gr/honeydrive, 2016. – abgerufen: 10.11.2016

[2] Kippo: *SSH Honeypot.* https://github.com/desaster/kippo, 2016. – abgerufen: 10.11.2016

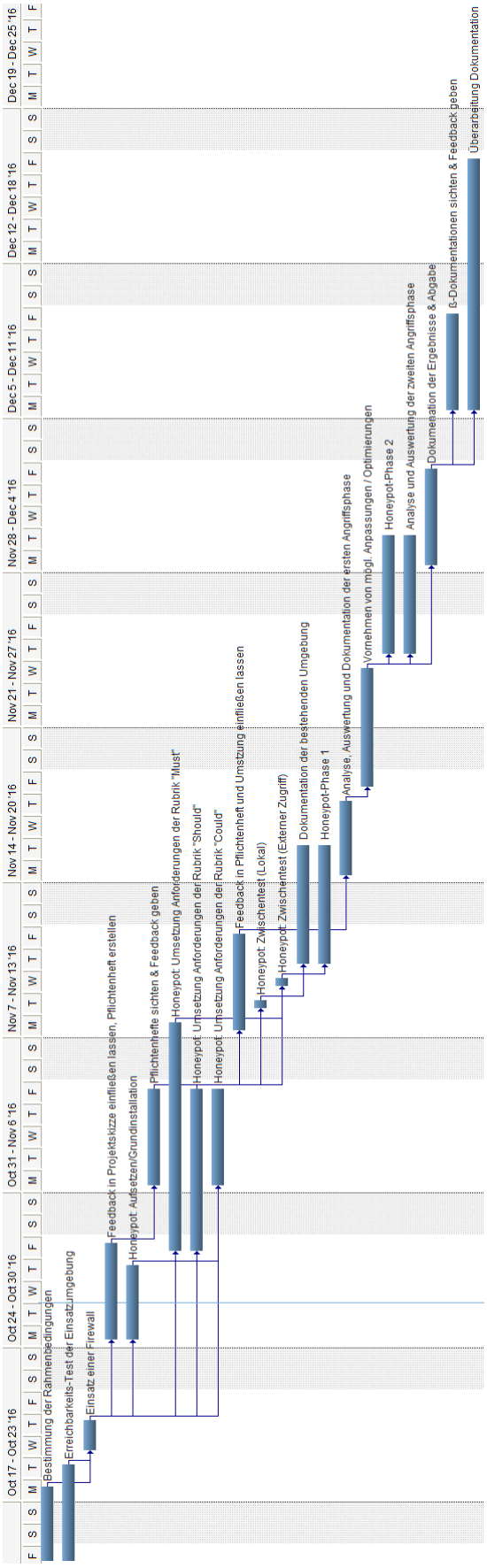
[3] Honeyd: *Developments of the Honeyd Virtual Honeypot.* http://www.honeyd.org/, 2016. – abgerufen: 10.11.2016

[4] Glastopf: *Web Application Honeypot.* http://glastopf.org, 2016. – abgerufen: 10.11.2016

A Anhang

Gantt-Diagramm

	Name	Duration	Start	Finish	Predecessors
1	Bestimmung der Rahmenbedingungen	2d	14/10/2016	17/10/2016	
2	Erreichbarkeits-Test der Einsatzumgebung	3d	14/10/2016	18/10/2016	
3	Einsatz einer Firewall	2d	19/10/2016	20/10/2016	1,2
4	Feedback in Projektskizze einfließen lassen, Pflichtenheft erstellen	5d	24/10/2016	28/10/2016	3
5	Honeypot: Aufsetzen/Grundinstallation	4d	24/10/2016	27/10/2016	3
6	Pflichtenhefte sichten & Feedback geben	5d	31/10/2016	04/11/2016	4
7	Honeypot: Umsetzung Anforderungen der Rubrik "Must"	7d	28/10/2016	07/11/2016	3,5
8	Honeypot: Umsetzung Anforderungen der Rubrik "Should"	6d	28/10/2016	04/11/2016	3,5
9	Honeypot: Umsetzung Anforderungen der Rubrik "Could"	5d	31/10/2016	04/11/2016	3,5
10	Feedback in Pflichtenheft und Umstzung einfließen lassen	5d	07/11/2016	11/11/2016	6
11	Honeypot: Zwischentest (Lokal)	1d	08/11/2016	08/11/2016	7,8
12	Honeypot: Zwischentest (Externer Zugriff)	1d	09/11/2016	09/11/2016	7,8,11
13	Dokumentation der bestehenden Umgebung	4d	10/11/2016	15/11/2016	11,12
14	Honeypot-Phase 1	4d	10/11/2016	15/11/2016	12
15	Analyse, Auswertung und Dokumentation der ersten Angriffsphase	4d	14/11/2016	17/11/2016	10
16	Vornehmen von mögl. Anpassungen / Optimierungen	4d	18/11/2016	23/11/2016	15
17	Honeypot-Phase 2	4d	24/11/2016	29/11/2016	16
18	Analyse und Auswertung der zweiten Angriffsphase	4d	24/11/2016	29/11/2016	16
19	Dokumenation der Ergebnisse & Abgabe	5d	28/11/2016	02/12/2016	16
20	ß-Dokumentationen sichten & Feedback geben	5d	05/12/2016	09/12/2016	19
21	Überarbeitung Dokumentation & Abgabe	10d	05/12/2016	16/12/2016	19


Oct 17 - Oct 23 '16
Oct 24 - Oct 30 '16
Oct 31 - Nov 6 '16
Nov 7 - Nov 13 '16
Nov 14 - Nov 20 '16
Nov 21 - Nov 27 '16
Nov 28 - Dec 4 '16
Dec 5 - Dec 11 '16
Dec 12 - Dec 18 '16
Dec 19 - Dec 25 '16
Bestimmung der Rahmenbedingungen
Erreichbarkeits-Test der Einsatzumgebung
Einsatz einer Firewall
Feedback in Projektskizze einfließen lassen, Pflichtenheft erstellen
Honeypot Aufsetzen/Grundinstallation
Pflichtenhefte sichten & Feedback geben
Honeypot Umsetzung Anforderungen der Rubrik "Must"
Honeypot Umsetzung Anforderungen der Rubrik "Should"
Honeypot Umsetzung Anforderungen der Rubrik "Could"
Feedback in Pflichtenheft und Umsetzung einfließen lassen
Honeypot Zwischentest (Lokal)
Honeypot Zwischentest (Externer Zugriff)
Dokumentation der bestehenden Umgebung
Honeypot-Phase 1
Analyse, Auswertung und Dokumentation der ersten Angriffsphase
Vornehmen von mögl. Anpassungen / Optimierungen
Honeypot-Phase 2
Analyse und Auswertung der zweiten Angriffsphase
Dokumentation der Ergebnisse & Abgabe
ß-Dokumentationen sichten & Feedback geben
Überarbeitung Dokumentation