HAW HAMBURG

Grundseminararbeit

Lukas Hettwer

Einblick in aktuelle Honeypotansätze

Betreuung durch: Prof. Dr. Kai von Luck / Prof. Dr. Jan Sudeikat
Eingereicht am: 28. Februar 2019

Fakultät Technik und Informatik
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Aktueller Stand zur IT-Sicherheit 2

2 Honeypot **2**
2.1 Einfluss auf die IT-Sicherheit 3
2.2 Eigenschaften und Typen von Honeypots 3
2.2.1 Honeypots im Produktionsumfeld 5
2.2.2 Research Honeypots 5
2.3 Schwierigkeiten mit Honeypots 6
2.4 Rechtliche Schwierigkeiten mit Honeypots 7

3 Beispiel an Honeypots **8**
3.1 Cowrie 8
3.2 BW-Pot 8
3.3 Dionaea 9
3.4 Thug 10

4 Zusammenfassung **10**

Selbstständigkeitserklärung **14**

Diese vorliegende Arbeit setzt sich theoretisch mit der Honeypot-Technologie auseinander und klassifizieren vier öffentliche zugängliche Honeypots. Dabei handelt es sich um den low interaction Honeypot Cowire, den high interaction Honeypot BW-Pot, um Dionaea, ein medium interaction Honeypot um Maleware zuerkennen und um den Honeypot-Client Thug. Ziel dieser Ausarbeitung ist es, das Konzept von Honeypots zu erklären und aktuelle öffentliche Honeypots vorzustellen.

Keywords: Honeypot, Cyberkriminalität, IT-Sicherheit

1 Einleitung

Durch die vermehrte Nutzung und Ausbaus des Internets entstehen im Gegenzug neue Gefahren. Ein Rechenzentrum im Internet kann durch einen Angreifer kompromittiert werden, ohne das die Betreiber dies bemerken können. Komplexe Sicherheitslücken, wie Heartbleed [26], Meltdown [15] oder Spectre [14] haben in der nahen Vergangenheit erneut gezeigt, dass Sicherheitskritische Bugs auch in hochprofessionelle Unternehmen, wie Amazon Web Services (AWS) oder Google Cloud, auftreten.

Eine Übernahme durch einen Angreifer ist prinzipiell immer möglich, doch durch verschiedene Sicherheitsmaßnahmen und Vorkehrungen kann die Wahrscheinlichkeit reduziert werden. Um einen Server im Internet vor der Infiltration zu schützen, muss verstanden werden, in welcher Form der Angreifer vorgeht. Aufgestellte Fallen, auch genannte Honeypots, zeichnen die Handlungen eines Hackers zum Zeitpunkt des Angriffs auf. Diese Aufzeichnungen tragen dazu bei, Angriffsmuster zu verstehen und Sicherheitsmechanismen zu entwicklen. Die Herausforderung besteht darin, dass der Honeypot das Verhalten des Angreifers aufzeichnet, ohne dabei selbst Teil des Angriffs zu werden.

Diese vorliegende Arbeit setzt sich theoretisch mit der Honeypot-Technologie auseinander und klassifizieren vier open-source Honeypots.

1.1 Aktueller Stand zur IT-Sicherheit

In der IT-Sicherheit werden Honeypots genutzt, um Studenten und Experten das vorgehen hinsichtlich von Angriffe verständlich zu machen [12] [20]. Es existieren besonders viele unterschiedliche Arten von Honeypots [17]. Besonders beliebt sind Honeypots in virtuellen Maschinen. Der Hypervisor ist eine geeignete Entwicklung, um die Umgebung sicher voneinander zu trennen. Das ist eine wichtige Voraussetzung, um Angreifer in einen Bereich des Systems zu beschränken. Bevorzugt wird die secure shell (SSH) imitiert [3] [21]. Das liegt daran, dass diese besonders gerne angegriffen werden und direkt Befehle ausführen können.

Mit dem Botnets Mirai rücken kleine leistungsschwache Maschinen in den Fokus der IT-Sicherheit [4] [19]. Um zu erforschen, wie Angreifer Armeen von IoT Geräten zusammen sammeln und steuern, werden globale IoT-Honeypotnetzwerke aufgebaut. In diesem Fall setzten die Forscher wenige IoT-Geräten ein, dafür viele verteilte Zugänge um den Durchsatz an Angriffe zu erhöhen [10]. Die klassischen virtuellen Maschinen (VM) haben nicht ausgedient, diese sind ein wichtiger Baustein in der Begrenzung von Systemen innerhalb eines Servers. VMs haben nicht mehr einen Honeypot Prozess laufen, sondern durch die Abgrenzungen der leichtgewichtigen Container sind mehre Honeypots auf eine VM möglich und reduzieren Komplexität und Betriebskosten. Container emulieren Linux- und Windowsdienste. [16]

2 Honeypot

In der IT-Sprache wird als Honeypot (Honigtopf) ein Sicherheitsmechanismus bezeichnet, welcher erkennt, wenn ein Informationssystem durch Unbefugte genutzt wird. Die Funktionalitäten sind vielfältig. Im Allgemeinen ist ein Honeypot eine Ressource des Informationssystems, welche aus Daten besteht, die in seinem Umfeld als legitim erscheinen. Tatsächlich sind Honeypot und die Daten isoliert, werden überwacht und haben keinen Wert für den Angreifer. Der Wert liegt darin, dass die Daten missbraucht werden. [8]

Ein Honeypot ist nicht für den rechtmäßigen Gebrauch bestimmt, daher ist jede Nutzung des Honeypots unrechtmäßig. Honeypots sind in der Lage aufkommende Bedrohungen frühzeitig zu erkennen, das Verhalten des Angreifers zu analysieren und (0-day) Exploits aufzudecken. [13]

2.1 Einfluss auf die IT-Sicherheit

In der IT-Sicherheit ist das Hauptziel Risiken zu reduzieren. Diese Risiken werden mit Gegenmaßnahmen reduziert. Wenn ein Risiko nicht einzudämmen ist, werden Pläne entworfen, die den Schaden durch das eintreten des Risikos verringern. Um eine Gefahr eines Cyberangriffes zu senken, bietet die IT-Sicherheit zahlreiche Verfahren an, wie z. B. der IT-Grundschutz Katalog [11] oder ISO-2700x [23], um systematisch Gefahren zu erkennen und zu beseitigen. Da aber all diese Verfahren das Problem haben, nur eine Teilmenge an Gefahren zu offenbaren und zu minimieren, wird neben präventive Mittel auch Erkennungsmechanismen von Angriffen eingesetzt.

> „Prevention is ideal, but detection is a must.“ [8]

Wird eine wichtige Ressource eines Unternehmens angegriffen, muss eine Komponente bereit sein, den Angriff so früh wie möglich zu erkennen. Dies muss unabhängig von der ausgenutzten Sicherheitslücke sein. Honeypots sind in der Lage diese zu erkennen und schlagen Alarm, um das Vorgehen frühzeitig zu unterbinden.

Um ein vollständiges Verständnis für das Risiko zu erhalten, sollte das Risikomanagement Honeypots einsetzen, um die derzeitigen Gefahren für die zu beschützenden Werte zu erkennen.

2.2 Eigenschaften und Typen von Honeypots

Anhand von Designkriterien können Honeypots in mindestens vier Kategorien klassifiziert werden: Pure Honeypots, low interaction Honeypots, medium interaction Honeypots und high interaction Honeypots [1] [17].

Pure Honeypots sind Produktionssysteme, wobei keine unterschiede zu einem vollwertigen System existieren. Sollte ein Angreifer den Honeypot angreifen, wird das durch eine Komponente im Netzwerk überwacht, die auf der Verbindung des Honeypots zum Netzwerk installiert wurde. Jedes Paket wird von dem Angreifer und dem Honeypot abgefangen und ausgewertet. Es muss keine weitere Software für den Honeypot installiert werden. [1]

Low interaction Honeypots simulieren nur die Dienste, die häufig von Angreifern angefordert werden, wie z. B. SSH, HTTP-Server oder RDCMan. Es gibt kein Betriebssystem für den Angreifer, mit dem er interagieren kann. Da die Anwendungen relativ wenig

Ressourcen verbrauchen, werden kleine virtuelle Maschinen mit den Anwendungen problemlos auf einem physischen System verwaltet und haben daher geringe Kosten. Die virtuellen Systeme haben eine kurze Antwortzeit und es wird weniger Code benötigt, was die Komplexität des virtuellen Systems reduziert. Obwohl das Risiko mit Honeypots minimiert ist, sind low interaction Honeypots sehr begrenzt. Low interaction Honeypot bieten sich an, Spammer zu analysieren und können auch als aktive Gegenmaßnahmen gegen Würmer eingesetzt werden. [1][17]

Medium interaction Honeypots sind anspruchsvoller als low interaction Honeypots. Das Betriebssystem ist nicht teil des Honeypots, beziehungsweise der Angreifer sollte nicht mit dem Betriebssystem interagieren. Die simulierten Dienste sind technisch tief greifender. Da die Komplexität steigt, ist es wahrscheinlicher, dass der Angreifer eine Sicherheitslücke entdeckt. Das Angriffsrisiko ist dennoch überschaubar. Dem Angreifer wird das System besser emuliert, deshalb wird der Angreifer länger auf dem System bleiben und komplexere Angriffe probieren. [17]

Ein high interaction Honeypots bietet die gleiche Umgebung an wie ein Produktionssystem. Diese bieten eine Vielzahl von Diensten und Dienstleistungen an, mit den ein Angreifer interagieren kann. Es sind die höchstentwickelten und komplexesten Honeypots. Diese bergen das größte Risiko, da der Angreifer mit dem Betriebssystem interagiert. Im Allgemeinen liefert ein high interaction Honeypots mehr Informationen über den Angreifer, als der low interaction Honeypot. Da Angreifer diese schwer als Honeypot identifizieren kann. Der Angreifer hat mehr Zeit auf dem System und offenbart dadurch sein Vorgehen. Das Ziel eines high interaction Honeypots ist es, dem Angreifer ein echtes Betriebssystem zur Verfügung zu stellen, mit dem er interagieren kann. Die Möglichkeiten, große Mengen an Informationen zu sammeln, sind daher bei dieser Art von Honeypot größer, da alle Aktionen protokolliert und analysiert werden. Sind die Informationen richtig eingesetzt, erhöhen diese deutlich die Sicherheit des Netzwerks. Dafür sind high interaction Honeypots teurer in der Wartung. Da der Angreifer über mehr Ressourcen verfügt, sollte ein Honeypot mit hoher Interaktion ständig überwacht werden, um sicherzustellen, dass er nicht zu einer Gefahr wird. [1] [17]

Ein Honeynet erweitert das Konzept der Honeypots um ein kontrolliertes Netzwerk an Pots. Ein Honeynet konzentriert sich auf das Überwachen aller Aktivitäten der Angreifer, speichert die Daten der Honeypots zentral ab und bietet die Möglichkeit, die Daten zu analysieren. [2]

2.2.1 Honeypots im Produktionsumfeld

Ein Produktions-Honeypot wird innerhalb der Umgebung eines Unternehmens verwendet, um das Unternehmen zu schützen und Risiken zu minimieren. Honeypots im Produktionsumfeld sind meistens einfacher zu bedienen, da diese wenige Funktionalitäten anbieten und nur begrenzte Informationen erfassen. Obwohl Angriffsmuster identifiziert werden, geben diese wenig Informationen über die Angreifer preis. Die Honeypots werden innerhalb des Produktionsnetzwerks mit anderen Produktionsservern platziert. Diese bilden das Produktionsnetzwerk des Unternehmens ab. Honeypots im Produktionsumfeld sind low interaction Honeypots, die wie zuvor schon beschrieben, leichter zu verwalten sind. [1] [17]

Wenn Server mit Millionen von Anfragen im Internet von einem Hacker angegriffen werden, ist das Ereignis schwierig zu erkennen. Befindet sich im Netzwerk der Server ein Honeypot, der sich von außen nicht von den anderen Servern unterscheidet, kann es sein, dass der Hacker diesen Server angreift. Er kann nicht erkennen, welcher der Server für ihn von Interesse ist und welcher der Honeypot ist. Der Honeypot hat keine Millionen von Anfragen, sondern nur die des Hackers. Der Angriff ist sofort erkannt und die schädliche Anfrage durch eine IP-Adresse ist identifiziert. Diese Informationen werden genutzt, um eine bessere Verteidigung aufzubauen. [8]

2.2.2 Research Honeypots

Research Honeypots dienen der Informationsbeschaffung und sind nicht direkt für die Sicherheit in einem Unternehmen verantwortlich. Diese werden verwendet, um Informationen über die allgemeinen Bedrohungen zu sammeln und zu erforschen. Mit diesen Informationen schützen sich Unternehmen besser vor den Risiken. Dabei wird die Art und Weise des Angreifers beim Eindringen in das System aufgezeichnet und untersucht, um Motive, Verhalten und Techniken zu verstehen. Es werden high interaction Honeypots eingesetzt, diese zeichnen länger und genau die Angriffe auf. Research Honeypots werden hauptsächlich von Forschungs-, Militär- und Regierungsorganisationen verwendet. [1] [8]

2.3 Schwierigkeiten mit Honeypots

Bei dem Einsatz von Honeypots muss mit Schwierigkeiten umgegangen werden. Eine Gefahr ist die Übernahme des Honeypots durch eines Angreifers. Ein Honeypot ist ebenfalls eine Applikation die durch den Menschen entwickelt wurde, daher ist es nicht auszuschließen, dass der Honeypot fehlerbehafteten Code ausführt. Es muss davon ausgegangen werden, dass Honeypots Schwachstellen besitzen. Bei einem low interaction Honeypot mag das Risiko einer Übernahme gering sein, da das Design dies nicht zulässt. Das drunterliegende Betriebsystem kann soweit beschnitten werden, sodass nur noch der Honeypot betrieben wird. Das Netzwerk um den Honeypot kann nur eingehende Verbindung erlauben und alles aus dem Honeypot blockieren.

Bei einem high interaction Honeypot ist eine Übernahme gefährlicher, denn dem Eindringling steht das komplette System zur Verfügung und der Angreifer ist in der Lage, Limitierungen, zum Beispiel ein Timer oder eine Ressourcenregulierung, zu deaktivieren. Die Gefahr besteht, dass die Überwachung des Honeypots manipuliert wird, sodass externe Server nichts vom Eindringen mitbekommen. Ein high interaction Honeypot wird sehr genau überwacht und den Informationen des Servers kann nicht ausschließlich vertraut werden. Externe Systeme müssen den Zugriff über das Netzwerk zusätzlich kontrollieren. [25]

Wenn ein Honeypot nicht übernommen wird, dann besteht die Gefahr, dass dieser enttarnt wird. Ist der Honeypot enttarnt, so weiß der Angreifer welches System er meiden sollte, um nicht Alarm zu schlagen. Der Honeypot ist zu diesem Zeitpunkt wertlos. Bei falschen Informationen durch den Angreifer macht der Honeypot die Lage schlimmer. Die Verteidigungsmechanismen konzentrieren sich auf die falschen Angriffe. Damit ein Honeypot nicht entdeckt wird, muss er sich genauso wie Viren weiter entwickeln, keine einzigartige Signatur besitzen und wenn er durch ein falsches Verhalten sich verrät, muss dies verbessert werden. [25]

Für den Fall, dass die Schwachstellen des Honeypots nicht ausgenutzt werden und dieser wird nicht enttarnt, heißt es nicht, dass das Projekt erfolgreich läuft. Es zeigt, dass kein Angreifer sich mit dem Honeypot beschäftigt. Ein research Honeypot hat als Hauptziel angegriffen zu werden, um Daten zu sammeln, daher muss der Honeypot nicht nur eine Internetverbindung haben, sondern eine prominente Position innerhalb des Netzes erhalten, sodass die interessanten Angreifer sich mit ihm beschäftigen. [25]

2.4 Rechtliche Schwierigkeiten mit Honeypots

Die rechtlichen Schwierigkeiten von Honeypots sind in drei Klassen einzuteilen: entrapment (Fallen stellen), privacy (Datenschutz) und liability (Haftung). Zusätzlich ist zu klären, welche Rechte gelten, da Cybercrime ein globales Problem ist. Deshalb muss das Gesetz des Standorts des Honeypots und des Angreifers, eingehalten werden. [22] [24]

Sollte der Honeypot genutzt werden um Straftaten aufzuzeichnen, muss geklärt werden, ob die Organisation rechtlich in der Lage ist, eine Falle zu stellen. Ein Honeypot provoziert mit seiner Anwesenheit Angriffe. Ohne diese Falle kann der Angreifer nicht die Straftat ausführen. Möglicherweise würde er diese Straftat in abgeänderten Form nicht begehen. Für den Fall, dass ein Angriff aufgezeichnet wird und die Organisation dazu berechtigt ist, muss forensisch die Beweise zusammengetragen werden, damit sie juristisch einzusetzen sind. [18] [22]

Bei dem Datenschutz ist es abhängig, welcher Typ von Honeypot eingesetzt wird. Ein low interaction Honeypot zeichnet weniger Daten auf die zu beschützen sind, im Vergleich zu einem high interaction Honeypot. Bei den Daten muss zwischen Inhalte und Aufzeichnungen von Aktionen unterschieden werden. Es muss geklärt werden, welche Daten erlaubt sind zu sammeln und wie mit den Daten umgegangen wird. Hier sind erneut die Herkunft des Angreifers und Ort des Honeypots zu berücksichtigen. Für den Fall, dass der Honeypot Daten aufzeichnet, die ein Einverständnis des Users benötigt, muss dieses durch ein Banner besorgt werden. Das steht mit dem Charakter und der Tarnung des Honeypots im Widerspruch. Für den Fall, dass Daten sammeln rechtlich ist, muss für die Sicherheit der Daten garantiert werden, was wieder eine Herausforderung im Zusammenhang mit Honeypots bürgt. [22]

Die Haftung ist keine strafrechtliche sondern eine zivilrechtliche Angelegenheit [24]. Bei jedem Service mit einer Internetverbindung besteht die Gefahr, dass dieser übernommen wird und Schaden bei Dritten verursacht. Das gilt auch bei Honeypots, dabei muss noch genauer dieses Risiko besprochen werden, da ein Honeypot vom Design her schon Angreifer einlädt. Ebenso steigt das Risiko zwischen low und high interaction Honeypots, da ein high interaction Honeypot schwieriger zu überwachen ist und noch mehr Schaden bei Dritten anrichten kann. [22]

3 Beispiel an Honeypots

In diesem Abschnitt wird beispielhaft für die Vielfalltigkeit von Honeypots vier davon besprochen. Die Honeypots wurden zufällig ausgewählt, wobei darauf geachtet wurde, dass sie sich nicht ähneln und aktuelle Commits bei GitHub haben.

3.1 Cowrie

Cowrie ist ein low interaction Honeypot der SSH/Telnet emuliert. Der Honeypot ist eine Weiterentwicklung des veralteten Honeypot Kippo. Die Anwendung Cowrie wird hinter dem Port 22 platziert und wartet dort auf anfragen durch SSH Clients. Der Service wird zuvor konfiguriert, sodass entweder nur bestimmte User mit Passwörter zugelassen werden oder jede Anfrage erfolgreich in den Honeypot eindringt. Der Honeypot selber zeichnet jede Interaktion mit diesem in einer Log Datei auf und bietet nach dem erfolgreichen einloggen weitere Möglichkeiten an. So kann der Angreifer folgende Befehle auf ein simuliertes System ausführen: sleep, sync, chattr, uname -r, dir, help, poweroff, jobs, halt, reboot, users, lscpu, killall, killall5, reset. Darauf aufbauend wird jeder Befehl, der nicht bekannt ist, mit einem positiven Rückgabewert beendet, sodass automatisierte Scripts nicht abbrechen und weiter ihr Vorgehen offenbaren. [6]

Da der Honeypot jeden User und jedes Passwort akzeptieren, ist spätestens nach dem zweiten erfolgreichen Anmeldeversuche die Tarnung des Honeypots obsolet. Da im inneren des Honeypots nur eine geringe Menge an Befehlen ausgeführt werden kann, wird der Angreifer schnell erkennen, dass sein Angriff wirkungslos ist und das Interesse daran verlieren. Da der SSH Server nur simuliert wird, ist der Honeypot auch ungeeignet Exploits wie der Heartbeat-Bug zu offenbaren. Der Honeypot eignet sich hervorragend um einfache Angriffsmuster zu erkennen und Studenten in der IT-Sicherheit für Angriffe im Netz zu sensibilisieren. [6]

3.2 BW-Pot

BW-Pot ist ein high interaction Honeypot der Web Applications anbietet. BW-Pot steh hierbei für Breakable Web Applications Honeypot. Die Entwickler dokumentieren seinen

Honeypot auf japanisch. Da der Honeypot nicht sehr komplex ist und der Google Übersetzer in der Lage ist die Dokumentation ins englische zu übersetzen, kann der Honeypot innerhalb dieser Arbeit dennoch besprochen werden. [9]

Der Honeypot bedient die Protokolle HTTP und HTTPS. Dabei wird ein Nginx Proxy Container aufgestellt. Dieser zeigt auf eine Wordpress Instanz, auf eine Apache Tomcat, auf eine phpMyAdmin Anwendung, auf einen PHP Server mit Zugang zu der Webshell und einen WOWHoneypot, welcher ein low interaction Web Application Honeypot ist. WOW steht für Welcome to Omotenashi Web Honeypot. Jeder dieser Instanzen und die zusätzlichen MySQL Datenbanken für Wordpress und phpMyAdmin laufen in eigenen Containern. [9]

Jede Anfrage die zu dem Nginx Proxy geht, wird gespiegelt, aufbereitet und persistiert. Der Nginx Proxy selbst schreibt Logs die ebenfalls gespeichert werden. Die Anwendungen unterhalb sind im original Zustand und zeichnen nichts auf. Der Angreifer kann mit diesen vollständig Interagieren. [9]

Dieser Honeypot ist ein high interaction Honeypot, da der Angreifer mit allen Systeme interagieren kann. Die Aufzeichnung des Angreifers geschieht auf dem Transportweg und schränkt diesen nicht ein. Die Entwickler vertrauen darauf, dass selbst wenn der Angreifer in die Anwendungen eindringt, nicht aus dem Container ausbrechen kann. Um eine Übernahme der Anwendungen nicht langfristig zu dulden, resettet der Honeypot seine Container nach 24 Stunden. Sollte ein Angreifer durch die Anwendung in die Container eindringen und aus diesem ausbrechen, kann dieser selbstverständliche den Wiederherstellungsmechanismus deaktivieren. Um aussagekräftige Informationen über den Angriff einzusammeln, muss dieses erhöhte Risiko eingegangen werden.

3.3 Dionaea

Bei Dionaea handel es sich um einen Service der in Python geschrieben ist. Dieser bietet auf unterschiedlichen Ports diverse Protokolle an, um darüber Maleware einzufangen. Es befinden sich keine echten Anwendungen der Protokolle hinter den Ports, sondern nur Emulierungen dieser. Diese simulieren die Protokolle auf minimaler Ebene. Folgende Protokolle unterstützt Dionaea: Black hole, EPMAP, FTP, HTTP, Memcached, Mirror, MQTT ,MSSQL, MySQL, PPTP, SIP, SMB, TFTP, UPNP. Sobald eine Maleware durch einer der Dienste eingefangen wurde, wird sie in einem besonderen Bereich gespeichert und dem Sicherheitsexperten per Link in einer E-Mail bereitgestellt. [7]

Von der Art und Weise handelt es sich hierbei um einen low interaction Honeypot, da dieser aber einen großen Bereich an unterschiedlichen Anwendunge bereitstellt, kann hierbei von einem medium interaction Honeypot gesprochen werden. Da der Honeypot darauf aus ist, Informationen über neue Maleware zu sammeln und nicht aktiv Alarm schlägt, ist der Einsatztort mehr in der Forschung einzuteilen. Das schließt natürlich nicht aus, das dieser in einem Firmennetzwerk eingesetzt werden kann, um Administratoren und deren Belgschaft für aktuelle Maleware zu sensibilisieren.

3.4 Thug

Bei Thug handelt es ich um einen Honeypot für die Clientseite, auch Honyclient genannt. Dieser baut auf einem hybriden statisch-dynamischen Analyseansatz auf und bietet eine DOM-Implementierung, die HTML, Events, Views und Style-Spezifikationen unterstützt. Thug nutzt die Google V8 Javascript-Engine um bösartigen Javascript-Code zu analysieren, um Shellcodes zu erkennen und diese zu emulieren. Thug kann als sehr geschwätziger Browser verstanden werden, der jeden Verstoß und jede Anomalie dokumentiert. Die Anwendung wartet nicht auf Angreifer, sondern fragt unter Aufsicht der Experten bei bösartigen Server nach und analysiert die Antwortpakete. [5]

Es handelt sich hierbei um ein low interaction Honeypot, da nur die Umgebung emultiert wird. Sollte ein Angreifer aus der Anwendung ausbrechen und mit dem Betriebsystem interagieren, handelt es sich um eine Schwachstelle. Da die Anwendung pro aktiv sein muss, um Angriffe aufzuzeichnen, handelt es sich um einen Honeypot für den Research Bereich.

4 Zusammenfassung

In dieser vorliegenden Ausarbeitung wurde das Konzept des Honeypots vorgestellt, in unterschiedlichen Stufen zwischen low interaction bis zum puren Honeypot eingeteilt und in den zwei Anwendungsfelder Research und Produktion gruppiert. Zusätzlichen wurden die Risiken und damit aufbauende die allgemeinen und rechtlichen Schwierigkeiten behandelt. Zum Abschluss wurden vier unterschiedliche Honeypots die aktuell weiter entwickelt werden, beschrieben und nach den Honeypot-Kriterien eingestuft.

Literatur

[1] Honeypot_(computing). (2018). – URL https://en.wikipedia.org/wiki/Honeypot_(computing)

[2] Abbasi, F. H. ; Harris, R. J.: Experiences with a Generation III virtual Honeynet. In: *2009 Australasian Telecommunication Networks and Applications Conference (ATNAC)*, Nov 2009, S. 1–6

[3] Alata, E. ; Nicomette, V. ; Kaaniche, M. ; Dacier, M. ; Herrb, M.: Lessons learned from the deployment of a high-interaction honeypot. In: *2006 Sixth European Dependable Computing Conference*, Oct 2006, S. 39–46

[4] Antonakakis, Manos ; April, Tim ; Bailey, Michael ; Bernhard, Matt ; Bursztein, Elie ; Cochran, Jaime ; Durumeric, Zakir ; Halderman, J. A. ; Invernizzi, Luca ; Kallitsis, Michalis ; Kumar, Deepak ; Lever, Chaz ; Ma, Zane ; Mason, Joshua ; Menscher, Damian ; Seaman, Chad ; Sullivan, Nick ; Thomas, Kurt ; Zhou, Yi: Understanding the Mirai Botnet. In: *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC : USENIX Association, 2017, S. 1093–1110. – URL https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis. – ISBN 978-1-931971-40-9

[5] buffer: Welcome to Thug's documentation! (2019). – URL https://buffer.github.io/thug/doc/index.html

[6] Cowire: Cowrie SSH/Telnet Honeypot. (2019). – URL https://github.com/cowrie/cowrie

[7] DinoTools: Welcome to dionaea's documentation! — dionaea 0.8.0 documentation. (2019). – URL https://dionaea.readthedocs.io/en/latest/index.html

[8] Eric Cole, Stephen N.: Honeypots: A Security Manager's Guide to Honeypots. (2018). – URL https://www.sans.edu/cyber-research/security-laboratory/article/honeypots-guide

[9] graneed: BW-Pot (Breakable Web applications honeyPot). (2019). – URL https://github.com/graneed/bwpot

[10] Guarnizo, Juan ; Tambe, Amit ; Bhunia, Suman S. ; Ochoa, Martín ; Tippenhauer, Nils O. ; Shabtai, Asaf ; Elovici, Yuval: SIPHON: Towards Scalable High-Interaction Physical Honeypots. In: *CoRR* abs/1701.02446 (2017). – URL http://arxiv.org/abs/1701.02446

[11] Informatik, Bundesamt für Sicherheit in der: It-Grundschutz. (2018). – URL https://www.bsi.bund.de/DE/Themen/ITGrundschutz/grundschutz.html

[12] Jones, Jeremiah K. ; Romney, Gordon W.: Honeynets: An Educational Resource for IT Security. In: *Proceedings of the 5th Conference on Information Technology Education*. New York, NY, USA : ACM, 2004 (CITC5 '04), S. 24–28. – URL http://doi.acm.org/10.1145/1029533.1029540. – ISBN 1-58113-936-5

[13] Kasza, Peter: Creating honeypots using Docker. (2015). – URL https://www.itinsight.hu/blog/posts/2015-05-04-creating-honeypots-using-docker.html

[14] Kocher, Paul ; Horn, Jann ; Fogh, Anders ; ; Genkin, Daniel ; Gruss, Daniel ; Haas, Werner ; Hamburg, Mike ; Lipp, Moritz ; Mangard, Stefan ; Prescher, Thomas ; Schwarz, Michael ; Yarom, Yuval: Spectre Attacks: Exploiting Speculative Execution. In: *40th IEEE Symposium on Security and Privacy (S&P'19)*, 2019

[15] Lipp, Moritz ; Schwarz, Michael ; Gruss, Daniel ; Prescher, Thomas ; Haas, Werner ; Fogh, Anders ; Horn, Jann ; Mangard, Stefan ; Kocher, Paul ; Genkin, Daniel ; Yarom, Yuval ; Hamburg, Mike: Meltdown: Reading Kernel Memory from User Space. In: *27th USENIX Security Symposium (USENIX Security 18)*, 2018

[16] Memari, N. ; Hashim, S. J. ; Samsudin, K.: Container based virtual honeynet for increased network security. In: *2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*, Feb 2015, S. 1–6

[17] Mokube, Iyatiti ; Adams, Michele: Honeypots: Concepts, approaches, and challenges. In: *in ACM-SE 45: Proceedings of the 45th Annual Southeast Regional Conference, 2007*, S. 321–326

[18] Nance, K. ; Ryan, D. J.: Legal Aspects of Digital Forensics: A Research Agenda. In: *2011 44th Hawaii International Conference on System Sciences*, Jan 2011, S. 1–6. – ISSN 1530-1605

[19] Pa, Yin Minn P. ; Suzuki, Shogo ; Yoshioka, Katsunari ; Matsumoto, Tsutomu ; Kasama, Takahiro ; Rossow, Christian: IoTPOT: Analysing the Rise of IoT Compromises. In: *9th USENIX Workshop on Offensive Technologies (WOOT 15)*. Washington, D.C. : USENIX Association, 2015. – URL https://www.usenix.org/conference/woot15/workshop-program/presentation/pa

[20] Sadasivam, Karthik ; Samudrala, Banuprasad ; Yang, T. A.: Design of Network Security Projects Using Honeypots, 2004

[21] Sentanoe, Stewart ; Taubmann, Benjamin ; Reiser, Hans P.: Virtual Machine Introspection Based SSH Honeypot. In: *Proceedings of the 4th Workshop on Security in Highly Connected IT Systems*. New York, NY, USA : ACM, 2017 (SHCIS '17), S. 13–18. – URL http://doi.acm.org/10.1145/3099012.3099016. – ISBN 978-1-4503-5271-0

[22] Sokol, Pavol ; Míšek, Jakub ; Husák, Martin: Honeypots and honeynets: issues of privacy. In: *EURASIP Journal on Information Security* 2017 (2017), Nr. 1, S. 4. – URL https://doi.org/10.1186/s13635-017-0057-4. ISBN 1687-417X

[23] Standardization, International O. for: ISO/IEC 27000 family - Information security management systems. (2018). – URL https://www.iso.org/isoiec-27001-information-security.html

[24] Symantec: Honeypots: Are They Illegal? (2003). – URL https://www.symantec.com/connect/articles/honeypots-are-they-illegal

[25] Symantec: Problems and Challenges with Honeypots. (2004). – URL https://www.symantec.com/connect/articles/problems-and-challenges-honeypots

[26] Synopsys, Inc. h.: Heartbleed Bug. (2018). – URL http://heartbleed.com/

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen."

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: _____

Vorname: _____

dass ich die vorliegende Grundseminararbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Einblick in aktuelle Honeypotansätze

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____	_____	_____
Ort	Datum	Unterschrift im Original