

Der Blast-o-Mat v4

Ein Ansatz zur automatischen Erkennung und Sperrung von Malware infizierten Computern

Jens Hektor, Jan Göbel

RWTH Aachen
Rechen- und Kommunikationszentrum

1 Einleitung

Das Internet hat sich zu einer Plattform für die verschiedensten Arten von Anwendungsapplikationen entwickelt. Galten früher Computer mehr als ein Ersatz für die Schreibmaschine oder den Taschenrechner, so verbinden sie heute Menschen auf der ganzen Welt. Heutzutage ist es möglich über das Internet Flüge zu buchen, sein Girokonto zu verwalten, mit Aktien zu handeln oder ein Auto zu kaufen und das alles mit einem handelsüblichen Personal Computer (PC). Aus diesen Gründen speichern selbst Heimcomputer mittlerweile eine Menge an sicherheitsrelevanten Daten, wie zum Beispiel Passwörter zu online Shops, Kreditkartennummern oder Personal Identification Numbers (PIN). Darüber hinaus sorgt der Fortschritt der Computerindustrie in immer kürzer werdenden Abständen für schnellere Rechner und Breitbandanbindungen in fast jedem Wohnzimmer. Ebenso schnell fallen auch die Preise der Internetdienstleister, die mit Flatrates dafür sorgen, dass jeder Rechner heute permanent online sein kann und das meist ohne dass der Anwender sich über die damit verbundenen Risiken bewusst ist. Die Tatsache, dass nun auch Heimcomputer wertvolle Informationen oder Ressourcen in Form von Rechenleistung zu bieten haben, macht sie zu einem immer interessanter werdenden Ziel für Kriminelle. In diesem Zusammenhang sind nicht ausschließlich sensitive Daten von Bedeutung, sondern auch die zur Verfügung stehende Bandbreite, die zum Versenden von SPAM oder Denial of Service Attacks eingesetzt werden kann. Darüber hinaus hat der Einsatz von sich automatisch verbreitender bösartiger Software (Malware) das Aufspüren und Kompromittieren von anfälligen Computern stark vereinfacht, so dass nicht mehr nur wenige Rechner betroffen sind, sondern mehrere tausend in sehr kurzer Zeit.

Deshalb verlangt die derzeitige Situation nach mehr aktiven und vorbeugenden Maßnahmen um infizierte oder kompromittierte Rechner daran zu hindern, weitere Computer im Netzwerk zu attackieren. Darüber hinaus muss verhindert werden, dass sensitive Daten von kontaminierten Rechnern an Dritte übermittelt werden.

2 Blast-o-Mat

2.1 Einleitung und Überblick

Der *Blast-o-Mat* ist ein verteiltes Intrusion Detection and Reaction System (IDSR), welches an der RWTH Aachen entwickelt wurde um die Ausbreitung von Malware innerhalb des Universitätsnetzwerkes zu minimieren und zwar möglichst ohne manuelle Interaktion.

Die Software ist nun seit fast vier Jahren kontinuierlich verbessert und weiterentwickelt worden. Eine erste Version erschien im Jahr 2003 [Hek06] zu Zeiten der ersten großen Wurm-Wellen, wie Nimda [Uni01] oder Blaster [Wik03]. Letzterer trug schließlich auch zur Namensgebung bei. Diese anfänglichen Versionen dienten der Unterstützung des Administrators beim Informieren und Sperren von infizierten Rechnern und beinhalteten nur eine Sensorik zur Erkennung von Rechnern die nach offenen Ports scannten.

Im Zuge der Weiterentwicklung wurde der Blast-o-Mat 2006 komplett überarbeitet [Göb06], so dass nun auch mehrere verschiedene Sensoren im Einsatz sind. Darüber hinaus wurde die einzelnen Bereiche logisch aufgeteilt und ein verteiltes System konstruiert. Zwei neue Intrusion Detection Sensoren sind hinzugekommen: ein low-interaction *Honeypot* [Wik] mit dem Namen *Nepenthes* [Tea05] und ein weiterer Sensor zur Erkennung von Spam verschickenden Rechnern im Netzwerk.

Durch den Einsatz von Honeypots ist es möglich die Anzahl von Fehlalarmen drastisch zu reduzieren, da hier nur dann Alarm geschlagen wird, wenn tatsächlich ein Angriff stattgefunden hat. *Nepenthes* simuliert eine Reihe von Diensten mit bekannten Schwachstellen. Wird eine dieser Schwachstellen ausgenutzt wird eine Warnmeldung an den Blast-o-Mat geschickt um weitere Maßnahmen einzuleiten. Einziger Nachteil dieser Methode: der Honeypot muss die für einen Angriff genutzte Schwachstelle simulieren, um einen Angriff überhaupt als solchen zu erkennen. Deshalb lassen sich so nur bekannte Angriffe detektieren. Aus diesem Grund verfügt der Blast-o-Mat über zwei weitere Sensoren: den bereits aus den früheren Versionen bekannten PortScan Sensor und den eingangs bereits erwähnten Spam Sensor.

Dadurch deckt der Blast-o-Mat zwei Haupteinfallstore für die massive Verbreitung von Malware ab: die Verbreitung von und durch E-Mail (Spam Sensor) und die Verbreitung durch Ausnutzen bekannter Schwachstellen (*Nepenthes* und PortScan Sensor).

Abbildung 1 zeigt einen möglichen Aufbau des verteilten IDSR. Der Blast-Sniffer ist ein Programm welches Verbindungsinformationen von einem Netzwerkinterface liest und in eine zentrale Datenbank speichert. Der separate Sniffer wird eingesetzt um die Blast-PortScan und Blast-SpamDet Sensoren mit Verbindungsdaten zu versorgen. Da beide Sensoren auf denselben Daten operieren wurde das Sammeln der Daten kurzerhand ausgelagert. Zusätzlich erlaubt diese Variante das Sammeln von Informationen von verschiedenen Rechnern, falls kein zentraler Router zur Verfügung steht.

In den nachfolgenden Abschnitten beschreiben wir die einzelnen Sensoren im Detail.

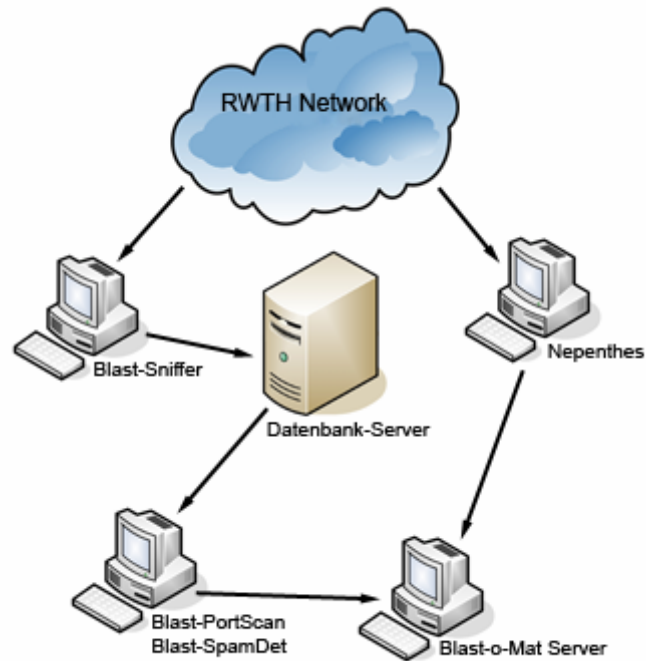


Abbildung 1: Schematischer Aufbau des verteilten Blast-o-Mat Systems

2.2 Die Blast-o-Mat Sensoren

2.2.1 Blast-Sniffer

Der *Blast-Sniffer* dient als Eingabequelle für die Blast-PortScan und Blast-SpamDet Intrusion Sensoren. Da diese dieselben Daten als Grundlage verwenden, ist es sinnvoll für beide eine gemeinsame Eingabequelle bereitzustellen. Der Blast-Sniffer lauscht an einem gewählten Netzwerkinterface und speichert alle TCP SYN Pakete in eine zentrale MySQL Datenbank. TCP SYN Pakete werden immer dann generiert, wenn ein Rechner versucht eine Verbindung mit einem anderen Computer herzustellen [Tho00], auch wenn die Verbindung nicht zustande kommt. Diese Pakete sind Teil des Drei-Wege-Handshake für TCP Verbindungen. Da die aktuellen Blast-o-Mat Sensoren keinen Paketinhalt untersuchen, um kompromittierte Rechner aufzudecken, reichen diese einleitenden Verbindungsdaten völlig aus. Um die Anzahl der Daten, die gespeichert werden, weiter einzuschränken, werden nur die Netzwerkpakete mitgelesen, die einen bestimmten Zielpport haben.

Zusätzlich zur Wahl des Interfaces und der Liste der Zielpports, lässt sich auch noch eine Whitelist konfigurieren, um von vornherein bestimmte Rechner oder ganze Teilnetze vom Mitschneiden der Netzwerkdaten auszuschließen.

Für jedes Netzwerkinterface, welches in der Konfigurationsdatei angegeben ist, startet der Blast-Sniffer einen neuen Thread. Dieser Thread erzeugt eine Instanz des Software Tools `tcpdump` [JLM] mit den entsprechenden Parametern, um die gewünschten Ports und nur TCP SYN Pakete zu extrahieren. Jedes empfangene Paket wird in seine Bestandteile, Quell-IP Adresse, Quellport, Ziel-IP Adresse, Zielport und Zeitstempel zerlegt. Diese Daten werden anschließend an den MySQL Server "übermittelt und in einer entsprechenden Tabelle gespeichert. Um belegten Speicherplatz wieder freizugeben, wird jede Stunde ein Aufräum-Prozess gestartet, der nicht mehr benötigte Datenbank Einträge löscht. Derzeit werden Einträge also nur eine Stunde lang aufbewahrt und dann entfernt, da die Daten nach der Auswertung durch den PortScan oder Spam Sensor nicht mehr weiter benötigt werden.

2.2.2 Blast-PortScan

Der *Blast-PortScan* Sensor analysiert die Netzwerk Daten, die vom Blast-Sniffer in einer zentralen Datenbank abgelegt werden. Die Aufgabe dieses Programms ist es, Rechner ausfindig zu machen die massiv nach Computern mit bestimmten offenen Ports suchen. Jedes sich automatisch verbreitende Malware Programm benutzt bestimmte Ports um neue Opfer zu finden und anschließend zu infizieren und auch bestimmte Backdoor Ports, um sich selbst auf einen neu infizierten Rechner nachzuladen. Daher kann das Suchen nach bestimmten Ports als ein deutliches Anzeichen eines kompromittierten Rechners angesehen werden.

Um eben solche Rechner ausfindig zu machen, durchsucht der Blast-PortScan Sensor die Datenbank in regelmäßigen Abständen nach Einträgen die bestimmte Zielports haben und noch innerhalb eines konfigurierten Zeitintervalls liegen. Diese Einträge werden so zusammengefasst, dass für jeden Rechner, von dem die Verbindungen ausgegangen sind, ersichtlich ist wie viele unterschiedliche Zielcomputer dieser kontaktiert hat. Sobald die Anzahl Zielrechner einen bestimmten Schwellwert "übersteigt wird eine Warnmeldung generiert und an den Blast-Server geschickt. Die Erkennung durch den PortScan Sensor erfolgt derzeit in drei Schritten und benutzt zusätzliche Datenbank Tabellen um Zwischenergebnisse speichern zu können. Im ersten Schritt werden alle Einträge gesammelt mit einem Zielport innerhalb der von uns überwachten Menge und einem Zeitstempel der nicht älter ist als der von uns vorgegebene Wert. Um doppelte Einträge zu vermeiden werden die gefundenen Zeilen anhand der Ursprungs IP Adresse, dem Zielport und der Ziel-IP zusammengefasst. Ausgehend davon werden nun alle Einträge gesammelt, deren Anzahl von Zielcomputern den vorgegebenen Schwellwert überschreitet. Im letzten Schritt werden noch einige statistische Daten gesammelt, wie zum Beispiel die Anzahl verschiedener Ports nach denen gesucht worden ist. Als Ergebnis erhalten wir eine Tabelle in der jeder Eintrag einen Rechner darstellt, der massiv andere Maschinen nach offenen Ports scannt. Zusätzlich enthalten die Einträge noch Informationen darüber wie viele Pakete versendet worden sind, wie viele Zielrechner kontaktiert worden sind und nach wie vielen verschiedenen Ports gesucht worden ist. Anschließend wird für jeden Eintrag eine Warnmeldung an den Blast-Server gesendet, der dann das weitere Vorgehen bestimmt.

Um den Blast-PortScan Sensor an das jeweilige Netzwerk anzupassen gibt es eine Reihe von Konfigurationsparametern, von denen einige nachfolgend erläutert werden. Derzeit überwachen wir an der RWTH Aachen mit diesem Sensor 14 verschiedene Ports. Der "WaitTime" Parameter ist auf 3 Minuten gestellt und "TargetThreshold" hat der Wert 50. Das heißt der Sensor prüft alle drei Minuten die Datenbank und filtert alle Einträge heraus die nicht älter als drei Minuten sind. Ein infizierter Rechner muss also innerhalb von drei Minuten mehr als 50 verschiedene Computer auf mindestens einem der entsprechenden Ports kontaktieren, um von unserem System als Portscanner erkannt zu werden. Beide Werte sind von der Vorgänger Version des Blast-o-Mat übernommen worden, da sie sich über drei Jahre bewährt haben. Fehlalarme sind bis jetzt nicht wirklich ein Thema gewesen. Zwar gab es auch schon Benachrichtigungen die nicht auf eine Kompromittierung zurückzuführen waren, sondern auf falsch konfigurierte Software, aber diese Ausnahmen bilden eine nicht weiter nennenswerte Minderheit.

2.2.3 Blast-SpamDet

Der *Blast-SpamDet* Sensor basiert auf dem gleichen Prinzip wie der Blast-PortScan Sensor, berücksichtigt aber ausschließlich Rechner, die Port 25 kontaktieren, also Mailserver. Genauso wie der Blast-PortScan Sensor werden die Anzahl der Verbindungsversuche pro Rechner mit einem vorgegebenen Schwellwert verglichen und bei Überschreiten des selbigen wird ein Alarm ausgelöst. Um zu vermeiden, dass fälschlicherweise legitime Mailserver als Spammer verdächtig werden, verfügt auch der Blast-SpamDet Sensor über so genannte Whitelisten, die es einem erlauben, bestimmte IP Adressen von der Erkennung auszuschließen.

Der größte Unterschied gegenüber den anderen Sensoren ist, dass Blast-SpamDet nur eine "Vermutung" an den Blast-Server sendet. Im Normalfall führen eingehende Warnmeldungen am Blast-Server direkt zur Einleitung von definierten Gegenmaßnahmen. In diesem Fall allerdings wird ein weiteres externes Skript vom Server ausgeführt, mit dessen Hilfe der Netzwerkverkehr des verdächtigten Rechners für einen gewissen Zeitraum mitgeschnitten wird. Aus diesen Daten werden alle im eMail Verkehr verwendeten Absenderadressen herausgefiltert und gesammelt. Am Ende entscheidet die Anzahl verschiedener Absenderadressen darüber ob der betroffene Rechner als Spammer klassifiziert wird oder nicht.

Der Grund, warum dieser Zwischenschritt beim Blast-Server durchgeführt wird, und nicht schon direkt beim Sensor ist, dass der Sensor ein einzelner Prozess ist und daher für die Dauer des Mitschneidens der Absenderadressen blockiert wäre. Eine gleichzeitige Behandlung weiterer Spammer wäre also nicht möglich. Da der Server für jede eingehende Warnmeldung einen eigenen Thread erzeugt können hier durchaus simultan weitere Vorfälle bearbeitet werden. Außerdem wurde, um den verteilten Ansatz zu unterstützen, ein externes Skript geschrieben, welches das sammeln der Absenderadressen unternimmt. Dies ermöglicht es, den Blast-Server auf einem anderen Rechner laufen zu lassen als dem, der Zugriff auf das entsprechende Netzwerk hat.

```
tcpdump -l -nnn -i eth0 -q -A -tt -s300 -c50 '( dst port 25 &&
tcp[20:4] = 0x4D41494C && src host 134.130.xxx.xxx)'
```

Abbildung 2: Tcpdump Kommandozeile

Für das Mitschneiden und Extrahieren der Absenderadressen wird das Programm tcpdump mit den in Abbildung 2 dargestellten Parametern verwendet. Der wohl wichtigste Parameter hier ist `tcp[20:4] = 0x4D41494C`, denn er veranlasst tcpdump dazu nur solche Pakete aus dem Datenstrom zu extrahieren, in denen das Wort "MAIL" vorkommt. Da das Simple Mail Transport Protokoll (SMTP) vorsieht, dass Absenderadressen mit dem Schlüsselwort "MAIL FROM:" einzuleiten sind, liefert uns tcpdump genau die richtigen Zeilen aus den Netzwerkpaketen. Die Absenderadressen werden dann mit Hilfe eines regulären Ausdrucks aus den so gewonnenen Zeilen extrahiert und an den Blast-Server zurückgegeben. Das externe Skript sammelt mit obigen Parametern maximal 50 verschiedene E-Mail Adressen und läuft nicht länger als 40 Sekunden. Die derzeitigen Blast-Server Einstellungen sehen vor, dass ein Rechner, der mehr als 10 verschiedene Absenderadressen verwendet, eindeutig als Spammer klassifiziert werden kann.

Bekannte reguläre Mailserver stehen in einer Whiteliste und sind daher von der Spamererkennung ausgenommen.

2.2.4 Nepenthes

Nepenthes ist ein low-interaction Honeypot der im Juni 2005 veröffentlicht worden ist und momentan von der *MWCollect Development Crew* weiterentwickelt wird. Ein lowinteraction Honeypot ist ein Rechner der mit Hilfe von Software vorgibt viele verschiedene Schwachstellen zu besitzen. Nepenthes simuliert also die Existenz von Diensten, um möglichst viele Angreifer anzulocken. Insbesondere sich automatisch verbreitende Malware wie Würmer und Bots liegen im Visier von Nepenthes. Das Ziel ist eben diese bösartige Software zu sammeln, um sie anschließend weiter untersuchen zu können.

Um mit der ständig wachsenden Anzahl von Schwachstellen, die von neuer Malware ausgenutzt wird, mithalten zu können ist Nepenthes modular aufgebaut. Jede simulierte Schwachstelle wird als ein eigenständiges Modul dargestellt, wodurch es möglich ist beliebige neue Schwachstellen-Module hinzuzufügen. Dies ermöglicht es, Nepenthes als Intrusion Sensor den aktuellen Bedrohungen anzupassen.

Bei jedem erfolgreichen Angriff, bzw. der Ausnutzung einer Schwachstelle wird von Nepenthes ein Ereignis ausgelöst, das an bestimmte Module der Software weitergereicht wird. Zu diesem Ereignis kann man auch eigene Module registrieren und so mit entsprechenden Maßnahmen auf Angriffe reagieren. Diese ereignisgesteuerte Benachrichtigung haben wir uns zu Nutze gemacht, um Nepenthes als Intrusion Sensor in unser verteiltes Blast-o-Mat System zu integrieren. Zu diesem Zweck haben wir das *log-blastomat* Modul geschrieben, das bei einem Angriff auf den low-interaction Honeypot eine XML Nachricht an den Blast-o-Mat Server weiterleitet. Als Protokoll für den Datentransfer zwischen dem Blast- Server und log-blastomat benutzen wir UDP. Da

wir keine Daten vom Server zurück an den Sensor schicken müssen und der Sensor nicht blockiert werden soll falls eine Verbindung nicht zu Stande kommt ist UDP die bevorzugte Wahl. Der Nachteil ist, dass es durchaus möglich ist, dass Pakete verloren gehen können. Im schlimmsten Fall würde das zu einem unerkannten Eindringling führen. Da aber infizierte Rechner mehr als einmal aktiv sind und wir nicht nur einen Nepenthes Sensor betreiben sondern eine ganze Reihe ist dieses Risiko sehr gering. Für den Fall, dass der Blast-Server gar nicht mehr erreichbar sein sollte, werden Vorfälle von den Sensoren auch zusätzlich auf die lokale Festplatte geschrieben.

Sobald eines der Nepenthes Schwachstellenmodule erfolgreich getriggert wird, sammelt log-blastomat einige Daten über den Angreifer, die Nepenthes den registrierten Ereignismodulen zur Verfügung stellt. Zu diesen Daten gehört die IP Adresse des Angreifers, der genaue Zeitpunkt des Angriffs, der attackierte Port und der Name der ausgenutzten Schwachstelle. Diese Informationen werden anschließend in eine XML konforme Nachricht gebracht und zur weiteren Bearbeitung an den Blast-Server gesendet.

Eine ausführliche Beschreibung von Nepenthes und seiner Funktionsweise findet sich in [BK+06].

2.3 Der Blast-o-Mat Server

Der *Blast-Server* bildet das Herzstück der Blast-o-Mat Software. Hier laufen alle gesammelten Informationen der Intrusion Sensoren zu einzelnen Vorfällen zusammen. Der Server lauscht an einem vorkonfigurierten Port und nimmt UDP Pakete der Sensoren zur Weiterverarbeitung entgegen. Für jeden eingehenden Zwischenfall wird ein neuer Thread erzeugt, der sich dann ausschließlich mit der Bearbeitung dieses Falls beschäftigt. Dadurch ist es möglich mehrere Fälle gleichzeitig zu bearbeiten und im Falle eines Fehlers wird nur der betroffene Thread beendet, aber der Server läuft ohne Einschränkung weiter.

Die gängigsten Parameter lassen sich direkt über eine Konfigurationsdatei anpassen. Dazu gehört zum Beispiel die IP Adresse und der Port auf dem der Blast-Server Daten der Sensoren entgegen nehmen soll. Um Missbrauch vorzubeugen, müssen die IP Adressen der Intrusion Sensoren in der Konfigurationsdatei hinterlegt sein, denn der Server bearbeitet nur eingehende Nachrichten von autorisierten IP Adressen. Darüber hinaus sind alle eingehenden Nachrichten mit einem Hash versehen, der mit Hilfe eines shared secrets erzeugt wird. Durch diese Methoden soll verhindert werden, dass Fremde manipulierte Daten einschleusen können.

Weitere Parameter die sich konfigurieren lassen, sind die Zeit in Minuten zwischen zwei Warnungsnachrichten, die Zeit bis zum Trennen der Verbindung eines verdächtigen Rechners und die Anzahl der Trennungen, bis ein Rechner permanent gesperrt wird. Außerdem ist es möglich, den Blast-o-Mat in einen so genannten "Just Warn" Modus zu schalten. In diesem Modus werden lediglich Warnungen versandt, aber keine Trennungen oder Sperrungen vorgenommen. Zusätzlich gibt es noch einen "Fallback" Mechanismus, in dem alle Empfänger eMail Adressen mit einer vorkonfigurierten

Adresse überschrieben werden. Dieser Modus ist insbesondere in Testphasen hilfreich, aber auch in Netzwerken wo es unmöglich ist automatisiert die eMail Adressen der verantwortlichen Personen ausfindig zu machen. Abschließend enthält die Konfigurationsdatei noch Einstellungen für den Mailserver, der benutzt werden soll, um die Blast-o-Mat eMails zu versenden.

Die Kommunikation der Intrusion Sensoren mit dem Blast-Server erfolgt ausschließlich über XML Nachrichten, die im Klartext über das Netzwerk versendet werden. Abbildung 3 zeigt ein Beispiel einer solchen XML Nachricht, die vom Blast-PortScan Sensor an den Blast-o-Mat Server gesendet worden ist.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE BlastEvent SYSTEM "xmlblast.dtd">
<BlastEvent>
  <Type>PortScan</Type>
  <IP>137.226.xxx.xxx</IP>
  <TStart>1153222133</TStart>
  <TEnd>1153222311</TEnd>
  <Proceed>Kick</Proceed>
  <Ports>139,445</Ports>
  <NumPackets>7295</NumPackets>
  <NumTargets>4520</NumTargets>
  <Hash>fbe4f233327b6922ede7ffdcc84baa74</Hash>
</BlastEvent>
```

Abbildung 3: XML Nachricht für einen Blast-o-Mat PortScan Event

Das erste XML Element "Type" entscheidet über den weiteren Umgang mit dem Vorfall. In unserem Beispiel handelt es sich um einen Portscan, und folglich wird der Server die Daten an die Routine für Portscans weiterreichen. Folgende Typen sind bisher definiert: PortScan, Exploit und Spammer. Durch diese Unterscheidung ist es möglich jeden Vorfall unterschiedlich zu behandeln. Zum Beispiel könnte es wünschenswert sein, Rechner die ein Exploit ausführen, also vom Nepenthes Sensor erfasst worden sind, direkt zu sperren, da hier keine Fehlalarme auftreten können. Auf der anderen Seite soll aber für Rechner die einen Portscan durchgeführt haben erst nach einer Abmahnung und einer kurzfristigen Trennung vom Netz der Zugang komplett gesperrt werden.

Die nachfolgenden Elemente der XML Nachricht beschreiben den Angreifer bzw. den infizierten Rechner. Zunächst wird die IP Adresse übermittelt gefolgt vom Zeitraum in dem die auffällige Aktivität stattgefunden hat. Der "Proceed" Wert ist ähnlich zum "JustWarn" Modus des Servers und gibt zusätzlich an wie weiter vorgegangen werden soll. Hier gibt es zurzeit die Möglichkeiten "Inform" und "Kick". Im ersten Fall werden nur Warnungen an betroffene Rechner verschickt, im zweiten Fall wird dann auch nach entsprechender Einstellung eine Trennung und schließlich die Sperrung eines infizierten Rechners eingeleitet. Dadurch ist es möglich, auch Sensoren für den Blast-o-Mat zu schreiben die nur Warnungen versenden, unabhängig davon wie die Server Einstellungen sind.

Die letzten Elemente dienen mehr statistischen Zwecken, um feststellen zu können welche Ports involviert waren, wie viele Pakete verschickt worden sind und welche Ziel

Rechner kontaktiert worden sind. Für Nachrichten vom Nepenthes Sensor gibt es auch noch ein optionales Element, welches den Namen der Schwachstelle beinhaltet, die ausgenutzt worden ist. Am Ende steht der für diese Nachricht erzeugte Hash.

Sobald eine XML Nachricht am Server eintrifft, prüft der Blast-Server die Gültigkeit der Absender-IP Adresse gegen die Liste in der Konfigurationsdatei und evaluiert den mitgeschickten Hash. Ungültige Nachrichten werden mit einer Bemerkung in den Log Dateien des Server verworfen. Bei gültigen Nachrichten werden alle Daten extrahiert und es wird eine kurze Zusammenfassung des Ereignisses erzeugt, die direkt ins Log geschrieben wird. Anschließend wird die dem Type entsprechende Funktion aufgerufen um mit der Behandlung des Vorfalls zu beginnen. Zu diesem Zweck gibt es zwei Unterklassen, die die nötigen Funktionen bereitstellen. Zum einen ist die die `AccountLookUp` Klasse und zum anderen die `BlastHelper` Klasse.

AccountLookUp Klasse Diese Klasse enthält alle Funktionen, die nötig sind um die Benutzerinformationen zu einer gegebenen IP Adresse zu erhalten. Das beinhaltet sowohl das entsprechende Teilnetz zu dem ein infizierter Rechner gehört als auch die eMail Adresse der verantwortlichen Person. Die `AccountLookUp` Klasse ist so aufgebaut, das man ausgehend vom Teilnetz unterschiedliche Methoden anwenden kann, um die eMail Adresse eines Verantwortlichen zu beziehen. Zum Beispiel könnte für Rechner, die zum Teilnetz "dialup" gehören, also VPN oder DSL Kunden sind, die Accounting Datenbank eines RADIUS Servers abgefragt werden. Im Fall von fest vergebenen IP Adresse könnte man einen LDAP Server abfragen. Durch diese Konzept ist es möglich, die Blast-o-Mat Software den gegebenen Netzwerkinfrastrukturen individuell anzupassen. Fest integriert ist derzeit nur eine Funktion zum Abfragen einer FreeRadius MySQL Datenbank, alle anderen Informationen werden über externe Scripte geregelt, die dem jeweiligen Netzwerk angepasst sind. Als Ergebnis liefert die `AccountLookUp` Klasse ein so genanntes Dictionary Objekt an den Blast-Server zurück, welches alle nötigen Informationen über den zu untersuchenden Rechner enthält.

Dadurch, dass alle Funktionen zum Herausfinden der Benutzerdaten in einer separaten Klasse untergebracht sind, lässt sich der Blast-o-Mat recht einfach an neue Netzwerkstrukturen anpassen. So ist es möglich, die ganze Klasse auszutauschen oder um neue Funktionen zu erweitern ohne die grundlegende Struktur des Blast-o-Mat Servers verändern zu müssen. Deswegen verfügt auch die `AccountLookUp` Klasse über eine eigene Konfigurationsdatei. Hier lassen sich unter anderem die Teilnetze deklarieren, für die man Benutzerdaten zur Verfügung hat und welche Rechner man vom Netzwerk trennen kann oder zur WhiteListe gehören und nicht beeinträchtigt werden dürfen.

BlastHelper Klasse Die `BlastHelper` Klasse dient der Aufbereitung der gesammelten Informationen zu einem bestimmten Vorfall. Hier werden zum Beispiel Einträge aus den Logfiles extrahiert, um sie ausgehenden eMails anzuhängen. Zusätzlich findet sich in dieser Klasse auch die gesamte Anbindung an die zentrale MySQL Datenbank. Eingehende Vorfälle werden in eine separate Tabelle zusammen mit allen zu diesem Vorfall gesammelten Information geschrieben. Bereits bestehende Einträge werden

aktualisiert. Diese Informationen dienen später wieder als Eingabe für die Weboberfläche des Blast-o-Mat und für statistische Zwecke.

2.4 Handhabung von infizierten Rechnern

Abschließend stellt sich die Frage, wie man nun mit einem infizierten Rechner umgehen soll. Am einfachsten wäre es sicherlich gar nichts zu unternehmen und das Problem einfach zu ignorieren. Dieses Verhalten würde aber im schlimmsten Fall zu noch mehr infizierten Rechnern führen, die durchaus auch andere Dinge durchführen können als bloss Portscans, so dass es auch schnell rechtliche Konsequenzen nachsichziehen kann. Zusätzlich führt der erhöhte Netzwerkverkehr zu möglichen Erreichbarkeitsproblemen von anderen lokalen Diensten wie zum Beispiel eMail oder DNS.

Die beste Lösung vom Standpunkt der Netzwerksicherheit ist es, einen infizierten Rechner sofort vom Rest des Netzwerks zu trennen. Zum einen verhindert dies eine Weiterverbreitung der Schadsoftware und Infizierung weiterer Rechner, zum anderen wird dadurch vermieden, dass sensitive Daten von betroffenen Computern weiter an Dritte übermittelt werden.

Die Sperrung eines kompromittierten Rechners kann an vielen Stellen geschehen, man sollte sich allerdings im Klaren darüber sein, dass nach Beseitigung des Problems auch alle Schritte wieder rückgängig gemacht werden müssen. Zu den verschiedenen Möglichkeiten gehört unter anderem das Abschalten des Switchports oder im Fall von fest vergebenen IP Adressen, das Sperren selbiger oder der zugehörigen MAC an einer Firewall. Bei Computern mit dynamischen IP Adressen ist es erforderlich diese vom Einwahlservers zu trennen und eine erneute Authentifizierung zu unterbinden, was sich durch setzen eines Flags in den Benutzerdaten einfach realisieren lässt.

Der Blast-o-Mat selber verfügt über keine integrierten Mechanismen einen Rechner vom Netz zu trennen oder dauerhaft zu sperren. Diese Tätigkeiten werden ausschließlich über externe Scripte realisiert, die der gegebenen Netzwerkstruktur angepasst sind.

2.4.1 Quarantäne Netzwerk

Ein anderer Ansatz im Gegensatz zum direkten Sperren eines infizierten Rechners, ist es selbigen in ein so genanntes Quarantäne Netzwerk umzuleiten. Obwohl dieses Vorgehen eine gewisse Infrastruktur voraussetzt, ist es sicherlich die beste Methode im Umgang mit kompromittierten Rechnern, da man so noch zusätzliche Informationen über das System sammeln kann ohne andere Computer im Netzwerk zu gefährden. Sofern die Möglichkeit besteht bis zu jedem Endgerät dedizierte VLANs einzusetzen, wäre es möglich, durch Wechseln des VLANs einen infizierten Rechner in ein separates Netzwerk zu verlagern.

Der Blast-o-Mat verwendet derzeit eine etwas weniger komplexe Vorgehensweise: mit Hilfe zweier externer Scripte werden Rechner, deren Netzwerkverkehr durch einen zentralen Router läuft, auf einen Quarantäne Webserver umgeleitet. Hierzu wird die

MAC Adresse des Rechners, der umgeleitet werden soll, ermittelt und anschließend werden durch den Einsatz von IPTables Regeln alle HTTP Anfragen auf die Adresse des Webservers umgeleitet. Hier erhält der Benutzer zusätzliche Information darüber warum sein Computer als kompromittiert gemeldet wurde. Dies ermöglicht es schneller mit dem betroffenen Benutzer in Kontakt zu treten, da man nicht sicher sein kann ob jeder seine eMails regelmäßig oder überhaupt liest.

Um einen infizierten Rechner umzuleiten sendet der Blast-Server die Benutzerkennung, die IP Adresse und den Zeitpunkt des Vorfalls an ein externes Skript auf einem zentralen Firewall Rechner. Dort wird dann zunächst, mittels durchsuchen der DHCP lease Datei, die MAC Adresse des auffälligen Rechners ermittelt und diese dann über eine entsprechende IPTables Regel umgeleitet. Zusätzlich speichert das Skript die Nutzerkennung zusammen mit der MAC Adresse in einer Datei, die beim Neustart der Firewall ausgelesen wird, um die Umleitungen aufrecht zu halten.

3 Ergebnisse

Insgesamt wurden seit März 2006 2.192 Vorfälle vom Blast-o-Mat bearbeitet, von denen allerdings nur 937 zu unterschiedlichen Rechnern gehörten. Diese Zahl lässt sich weiter aufteilen in 387 Rechner mit fester IP Adresse, also zum Beispiel Instituts- oder Wohnheimrechner. Die restlichen 550 Maschine kamen aus dem DialUp bzw. VPN Bereich. Hierzu zählen insbesondere Kunden aus dem Wireless Local Area Networks (WLAN).

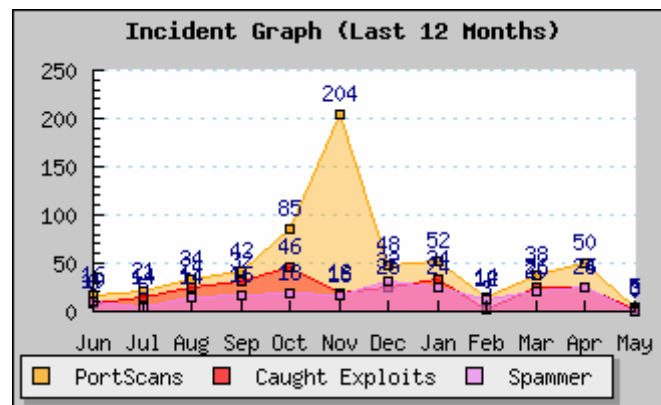


Abbildung 4: Ergebnisse des Blast-o-Mat in den letzten 12 Monaten

Abbildung 4 zeigt die Erkennungsraten der einzelnen Blast-o-Mat Intrusion Sensoren über die letzten zwölf Monate. Deutlich zu erkennen ist eine Wurmwelle im November 2006. Betroffen waren überwiegend Windows 2000 Systeme aus zwei verschiedenen Local Area Networks (LANs), so dass auch ein Sperren der infizierten Rechner keine Auswirkung auf die Verbreitung innerhalb des eigenen LANs hatte.

4 Zusammenfassung und Ausblick

Vorgestellt wurde ein Intrusion Detection and Reaction System, welches die wichtigen Aufgaben im Falle eines kompromittierten Rechners automatisch unternimmt: Die Benachrichtigung des für den Rechner verantwortlichen Administrators oder Benutzers, sowie die Sperrung der Maschine um eine Weiterverbreitung von Malware weitestgehend zu unterbinden. Darüber hinaus wurde der low-interaction Honeypot Nepenthes als neue Art von Intrusion Sensoren eingesetzt. Dieser Sensor bietet den großen Vorteil, keine Fehlalarme zu erzeugen und ermöglicht es zusätzlich auch tiefer gehende Informationen über die sich verbreitende Malware zu erfahren.

Für die Zukunft ist geplant den Blast-o-Mat dahingehend zu erweitern auch Intrusion Detection Message Exchange Format (IDMEF) Nachrichten [Kra] verarbeiten zu können. Dadurch wäre eine Integration in bereits bestehende Sicherheitsapplikationen einfacher zu gestalten. Darüber hinaus sind auch weitere Sensoren in der Planung, besonders im Bereich der Botnetzerkennung.

Literatur

- [BK+06] Paul Baecher, Markus Koetter, Thorsten Holz, Maximilian Dornseif, and Felix Freiling. The Nepenthes Platform: An Efficient Approach to Collect Malware. 2006.
- [Göb06] Jan Göbel. Advanced Honeynet-based Intrusion Detection. 2006. Diploma Thesis.
- [Hek06] Jens Hektor. Intrusion Detection, Notifying und Handling - Beitrag zu einem automatisierten Ansatz. 2006. 13.DFN Workshop.
- [JLM] Van Jacobson, Craig Leres, and Steven McCanne. tcpdump. <http://www.tcpdump.org> .
- [Kra] Robert Krauz. Intrusion Detection Message Exchange Format. <http://www-rnks.informatik.tu-cottbus.de/de/materials/ws2002seRecent/krauz.pdf>, Accessed: 2007.
- [Tea05] Nepenthes Development Team. Nepenthes: finest collection. 2005. <http://nepenthes.mwcollect.org/> .
- [Tho00] Thorsten Thormaehlen. Transmission Control Protocol (TCP). 2000. <http://www.thormahlen.de/diplhtml/node24.html> .
- [Uni01] Carnegie Mellon University. CERT Advisory CA-2001-26 Nimda Worm. 2001. <http://www.cert.org/advisories/CA-2001-26.html> .
- [Wik] Wikipedia. Honeypot. <http://de.wikipedia.org/wiki/Honeypot> .
- [Wik03] Wikipedia. W32.Blaster. 2003. <http://de.wikipedia.org/wiki/W32.Blaster> .