X-ARF: Network Abuse Reporting 2.0

Matthias Bräck, Sven Übelacker

Easterhegg 2011, Hamburg

Einleitung

- (SSH) Account Probes betreffen jeden Admin

  ```
  Feb 23 00:11:08 uebelhacker sshd[25108]: Invalid user bunny
  from 95.141.226.37
  ```

- Gegenmaßnahmen?
  ↪ andere Authentisierungwege wählen
  ↪ iptables-Regeln, nicht Standardports benutzen
  ↪ div. Tools (fail2ban, ..)
  ↪ bekämpfen oft nur Ursachen statt Symptome
- Hinter fast jedem Angriff steht ein bereits komprommitiertes System
  ↪ Wie kann man die Netz-Verantwortlichen kontaktieren?
  ↪ Und in welcher Form?

Einleitung

- Spam betrifft jeden User
- Gegenmaßnahmen?
  ↪ Spammarkierung
  ↪ Black-/Grey-/Whitelisting
  ↪ Netzblöcke/TLDs blockieren
  ↪ DKIM? Really?
  ↪ wieder nur Symptombekämpfung
- Wie kann man die Netz-Verantwortlichen in welcher Form kontaktieren?

Abuse Reporting Formate

- MMDEF (Malware Meta-Data Exchange Format)
- MAEC (Malware Attribute Enumeration and Characterization)
- IDMEF (Intrusion Detection Message Exchange Format)
- IODEF (Incident Object Description and Exchange Format)
- ARF (Abuse Feedback Reporting Format)

Idee und Motivation

- E-Mail ist eine fast immer vorhandene Infrastruktur
- frei, offen, community-driven
- Flexiblität und Einfachheit
- erweiterbar über definierte Schemata
- menschen- und maschinenlesbar
- Validierbarkeit

Einsatz

- DomainFactory: Reporting
- NetCologne: Honeypot Events
- Kabel BW: Honeypot Events
- GÉANT – Verbundprojekt der europäischen Forschungsnetze (NREN): X-ARF als Austauschformat
- Deutscher Dialogmarketing Verband (DDV)
  „Certified Senders Alliance“-Zertifizierung (Positivlisten) nur mit X-ARF seit 15.03.2011
  ↪ `http://www.certified-senders.eu`

X-ARF-Spezifikation

- E-Mailformat mit Headerfeldern und zwei/drei MIME-Parts
- Inhalt definiert über JSON-Schemata, derzeit:
  - abuse_login-attack
  - abuse_malware-attack
  - auth_login-attack
  - fraud
  - info
  - virus_bot

X-ARF-Header

- `X-ARF: yes` (nach RFC822)
- `Auto-Submitted: auto-generated` (nach RFC3834)
- `Content-Type: multipart/mixed`
- empfohlen:
  `Subject: abuse report about <source> - <date>`
  zum Beispiel:
  `Subject: abuse report about 95.141.226.37 - 2011-02-23`

X-ARF-Spezifikation

1. MIME-Part

- `Content-Type: text/plain`
- `charset=utf-8`
- rein menschenlesbarer Freitext
  ↪ sinnvoll für Empfänger, die kein X-ARF verstehen

2. MIME-Part

- menschen- und maschinenlesbares (und -validierbares) YAML-Format
- YAML-Inhalt (Schlüssel & Werte) hängt vom angewendeten X-ARF-JSON-Schema ab (`Schema-URL` in allen JSON-Schemata enthalten)
- YAML ist konvertierbar in JSON, dadurch können YAML-Daten mit JSON-Schemata validiert werden
- `Content-Type: text/plain`
- `charset=utf-8`
- `name="report.txt"`

2. MIME-Part: Pflichtfelder in jedem JSON-Schema

- `Reported-From:` Absender-E-Mail-Adresse
- `Category:` kann “abuse“, “fraud“, “auth“, “info“ oder “private“ sein
- `Report-Type:` wird je nach Kategorie von der X-ARF-Community definiert
- `User-Agent:` Name der generierenden Software
- `Report-ID:` [UNIXTIMESTAMP|randomnumber(4)]@domain.tld
- `Date:` Datum nach RFC3339
- `Source:` & `Source-Type:` legen den Verursacher fest (URL, E-Mail- oder IP-Adresse, ...)
- `Attachment:` Typ des Attachments (MIME-Typ oder “none“)
- `Schema-URL:` URL, wo das verwendete JSON-Schema zu finden ist

2. MIME-Part: Auszug aus JSON-Schema login-attack

```
{"description":"An abusive login-attack report",
 "type":"object",
 "properties":{
   "Reported-From":{
     "type":"string",
     "format":"email"
   },
   "Source":{
     "type":"string"
   },
   "Source-Type":{
     "type":"string",
     "enum":["ipv4","ipv6","ip-address"]
   },
   "Destination":{
     "type":"string",
     "optional":true,
     "requires":"Destination-Type"
   },
   "Destination-Type":{
     "type":"string",
     "enum":["ipv4","ipv6","ip-address"],
     "optional":true
   },
   "Attachment":{
     "type":"string",
     "enum":["text/plain"]
   },
```

3. MIME-Part

- `Content-Type` frei wählbar je nach JSON-Schema
- Kann (anonymisiertes) „Beweismaterial“ wie Logzeilen oder Malware-Samples enthalten
- Inhalt ist frei wählbar je nach JSON-Schema
- 3. MIME-Part kann je nach JSON-Schema auch optional sein

X-ARF-Beispiel: login-attack

1. MIME-Part: Freitext

X-ARF Beispiel mit JSON Schema abuse_login-attack 0.1.1

Freitext
Content-Transfer-Encoding: 7bit
Content-Type: text/plain; charset=utf8;

```
Dear Abuse Team,

this is an automated report for ip address 95.141.226.37 in
format „X-ARF“ generated on 2011-02-23 08:22:55 +0100

ip address 95.141.226.37 produced 19063 log lines, sample log
lines attached.

Regards,
Sven Uebelacker
```

X-ARF-Beispiel: login-attack

2. MIME-Part: YAML

report.txt
Content-Transfer-Encoding: 7bit
Content-Type: text/plain; charset=utf8; name="report.txt";

```
---
Reported-From: sven@uebelacker.net
Category: abuse
Report-Type: login-attack
Service: ssh
Port: 22
User-Agent: xarf-ssh-reporter.sh 2010-12-17
Report-ID: 1298400865131 5@uebelhacker.de
Date: Tue, 22 Feb 2011 19:54:25 +0100
Source: 95.141.226.37
Source-Type: ipv4
Attachment: text/plain
Schema-URL: http://www.x-arf.org/schema/abuse_login-attack_0.1.1.json
```

X-ARF-Beispiel: login-attack

3. MIME-Part: Logzeilen

logfile.log
Content-Transfer-Encoding: 7bit
Content-Type: text/plain; charset=utf8; name="logfile.log";

```
Feb 22 19:54:25 uebelhacker sshd[1965]: Did not receive
                    identification string from 95.141.226.37
Feb 22 21:05:44 uebelhacker sshd[10273]: Invalid user mat3
                    from 95.141.226.37
Feb 22 21:05:45 uebelhacker sshd[10275]: Invalid user matt4
                    from 95.141.226.37
-- MARK --
Feb 23 01:35:50 uebelhacker sshd[14658]: Invalid user venezia
                    from 95.141.226.37
Feb 23 01:35:51 uebelhacker sshd[14664]: Invalid user tsunami
                    from 95.141.226.37
```

Vorteile
für CSIRTs

- automatisiertes Abuse Handling
- aber auch Menschenlesbarkeit
- schnellere Vorfallsbearbeitung
- einfach zu parsen
- validierbar und erweiterbar

Vorteile

für Administratoren

- einfache Möglichkeit um Angriffe an verantwortliche Stellen zu melden
- Ursachen statt ausschließlich Symptome bekämpfen
- Lesbarkeit für Menschen
- „Your system has been compromised.“

Dilemma des Abuse-Kontakts

whois 95.141.226.37

```
inetnum:        95.141.224.0 - 95.141.231.255
netname:        CHEBNET
descr:          Internet Center Ltd
country:        RU
admin-c:        NoC117-RIPE
tech-c:         NoC117-RIPE
status:         ASSIGNED PA
mnt-by:         MNT-CHEBNET
source:         RIPE # Filtered

role:           NOC of CHEBNET
address:        Russian Federation,
address:        428000, Cheboksary, 139 Strelkovoy Divizyi street
abuse-mailbox: abuse@chebnet.com
```

Dilemma des Abuse-Kontakts

- whois-Abfragen beinhalten nicht immer abuse-Kontakte in automatisiert-lesbarer Form (Datenschutz? vgl. DENIC)
- Schwieriges Unterfangen: Wurde weltweit schon oft versucht, bisher existiert keine Lösung ↪ Penalties für AS-Betreiber?
- abusix.org Ansatz geht über DNS-Reverse-Lookups

abuse-contacts.abusix.org

```
dig +short 37.226.141.95.abuse-contacts.abusix.org TXT
"abuse@chebnet.com"
```

- Leider sind die Whois-Kontaktdaten nicht immer korrekt:

```
<abuse@chebnet.com>: host mx.yandex.ru[77.88.21.89] said:
550 5.7.1 No such user! (in reply to RCPT TO command)
```

Lösungsansatz für Abuse-Kontakte

anderer Versuch von Trusted Introducer: IRT-Objekt in der RIPE DB

IRT-Objekt im Whois für Einrichtungen im Deutschen Forschungsnetz

```
irt:             IRT-DFN-CERT
address:         DFN-CERT Services GmbH
address:         Sachsenstrasse 5
address:         20097 Hamburg
address:         Germany
phone:           +49 40 808077 590
fax-no:          +49 40 808077 556
abuse-mailbox:   dfncert@dfn-cert.de
signature:       PGPKEY-F12E83C7
encryption:      PGPKEY-F12E83C7
admin-c:         TI123-RIPE
tech-c:          TI123-RIPE
auth:            PGPKEY-F12E83C7
remarks:         This is a TI accredited CSIRT/CERT
remarks:         emergency phone number +49 40 808077 590
remarks:         timezone GMT+01 (GMT+02 with DST)
remarks:         https://www.trusted-introducer.org/teams/dfn-cert.html
irt-nfy:         dfncert@dfn-cert.de
mnt-by:          TRUSTED-INTRODUCER-MNT
source:          RIPE # Filtered
```

X-ARF: Infrastruktur & Vertrauenswürdigkeit

- Mailserverlast
  ↪ bisher pro Quell-IP-Adresse eine X-ARF-E-Mail
- DoS auf Trouble-Ticket-Systeme
  ↪ falls X-ARF-Meldungen direkt zu Ticketgenerierung führen
- E-Mail ist kein zuverlässiger Dienst
- Integrität/Vertrauenswürdigkeit einer Meldung
  ↪ Aufgabe des Abuse Handlers auf der Angreiferseite

Ausblick

- Bulk-Format für optimierten Datentransfer verschiedener Abuse-Typen (YAML-Listen?)
- Traffic Light Protocol (TLP): “white“, “green“, “amber“ oder “red“
- `Occurrences` für Einschätzung des Schweregrads der Vorkommnisse
- Glaubwürdigkeit erhöhen (PGP? DKIM? X.509?)
- Tobias Knecht (abusix): „Fighting Botnets Effectively“
  ↪ RIR news: Making abuse contacts mandatory in WHOIS
  ↪ APNIC introduced the mandatory IRT Object in November :-)
  ↪ AfriNIC is starting a Last Call for Proposal
  ↪ RIPE is talking and moving forward

Tools

- Python:
  `http://x-arfreporting.sourceforge.net`
- Ruby:
  `http://rubygems.org/gems/xarf`
- PHP:
  `http://www.blocklist.de/downloads/genxarf-php.tar.gz`
- Java Validator:
  `https://code.google.com/p/jxarfvalidator/`
- X-ARF-Validierung via PHP:
  `http://www.blocklist.de/downloads/validatexarf-php.tar.gz`
- X-ARF-Mail oder X-ARF-Yaml-Reports online überprüfen:
  `http://x-arf.org/validator.html`
- BASH ab 4.x: X-ARF SSH Probe Reporter:
  `ftp://ftp.dfn-cert.de/pub/tools/x-arf/`

Quellennachweise

- http://x-arf.org/
- https://github.com/xarf/xarf-Specification
- http://www.yaml.org/
- http://www.json.org/
- http://www.json-schema.org/