BİLGİ GÜVENLİĞİ AKADEMİSİ

www.bga.com.tr

Snort IPS (Intrusion Prevention System) Eğitimi

@2014

Örnek Eğitim Notu

bilgi@bga.com.tr

Snort Kuralları

IDS'lerde Kural/İmza Mantığı

- Kural mı imza mı?
  - İmza(signature)= trafik içerisinde "imza(xyz gibi)" arama
  - Kural(Rule)=İmza ve başka parçaları kontrol etme
  - Snort imza tabanlı değil, kural tabanlı bir IPS'dir!
- IDSler iki temel çalışma yöntemi
  - İmza tabanlı
  - Anormallik tabanlı
- İmzalar Vulnerability tabanlı olabilir
- İmzalar Exploit tabanlı olabilir

Kuralları Anlama ve Yorumlama

- Snort yapılandırmasının en önemli bileşenlerinden.
- Saldırı tespit sistemine ne yapacağını söyleyen bileşenlerden
  - Diğer bileşen Preprocessor(önişlemci)

Kural Çeşitleri

- Sourcefire kuralları
  - Ticari kurallar
  - Ücretsiz kurallar(30 gün gecikmeli?)
- SO kurallar
- BE kuralları
- Kendi geliştireceğiniz kurallar

Kural Sınıflandırmaları

- Tüm kurallar tek bir dosyadan alınmaz
- Saldırı kuralları çeşitli kategorilere bölünmüştür

```
Makefile.am               info.rules              smtp.rules
VRT-License.txt           local.rules             snmp.rules
attack-responses.rules    misc.rules              specific-threats.rules
backdoor.rules            multimedia.rules        spyware-put.rules
bad-traffic.rules         mysql.rules             sql.rules
cgi-bin.list              netbios.rules           telnet.rules
chat.rules                nntp.rules              tftp.rules
content-replace.rules     open-test.conf          virus.rules
ddos.rules                oracle.rules            voip.rules
deleted.rules             other-ids.rules         web-attacks.rules
dns.rules                 p2p.rules               web-cgi.rules
dos.rules                 policy.rules            web-client.rules
experimental.rules        pop2.rules              web-coldfusion.rules
exploit.rules             pop3.rules              web-frontpage.rules
finger.rules              porn.rules              web-iis.rules
ftp.rules                 rpc.rules               web-misc.rules
icmp-info.rules           rservices.rules         web-php.rules
icmp.rules                scan.rules              x11.rules
imap.rules                shellcode.rules
```

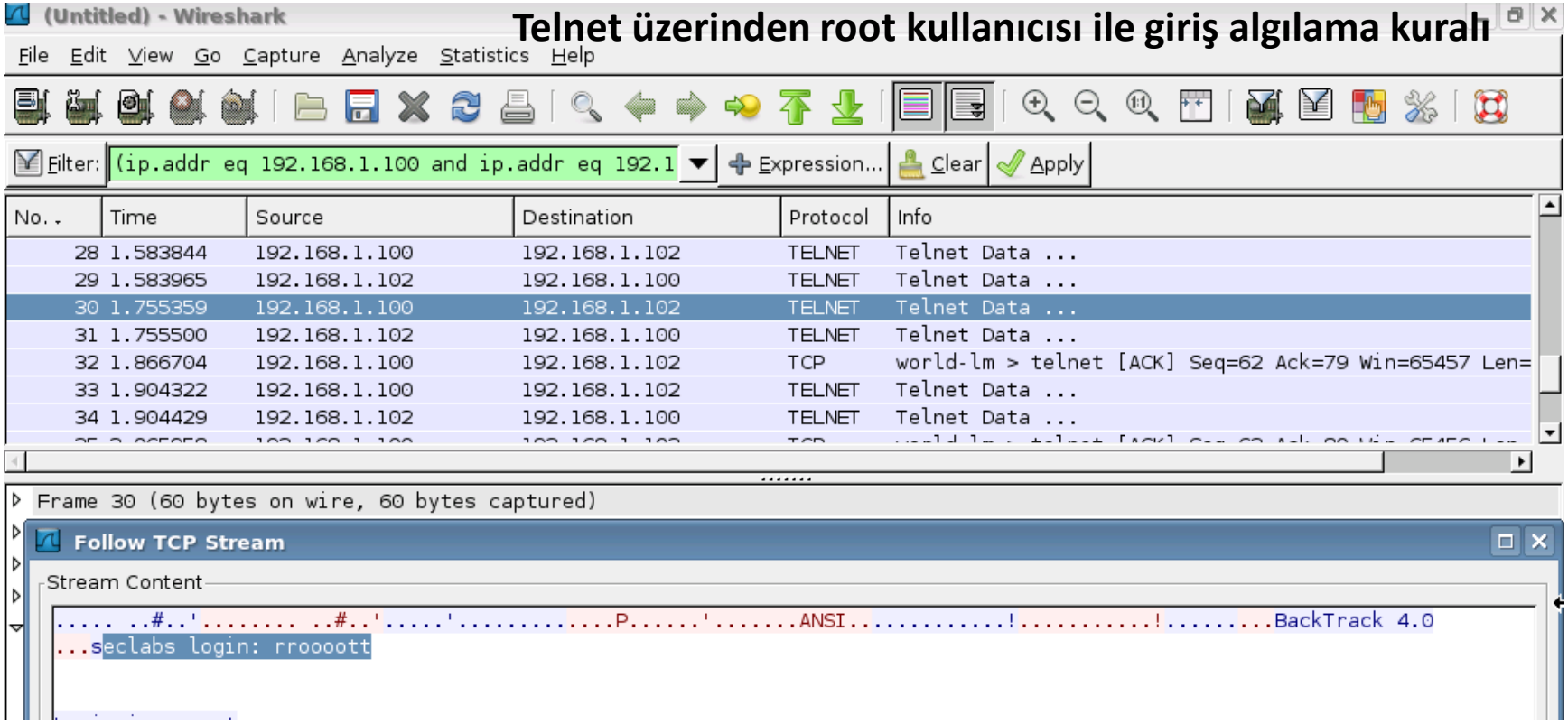
Kural Kategori İşlevleri

Kural Kategorisi	İşlevi
backdoor.rules	*Çeşitli trojanlar ve rootkitler tarafında oluşturulan trafiği saptamak için yazılmıştır.*
ddos.rules	*Bilinnen DDOS sa ldırılarını saptamak için kullanılır.*
Oracle.rules	*oracle veritabanı sunucusuna yapılabilecek saldırıları tespit eder.*
scan.rules	*Çeşitli ağ ve servis tarama araçlarının yaptığı taramaları tespti eder*
web-iis.rules	*Microsoft IIS'e yapılacak saldırılarıo tespit eder, eğer ağınızda IIS çalışıyorsa bu kural ailesinini aktif edilmesine gerek yoktur.*
p2p.rules	*P2P trafiği tespit etmek için kullanılır*

IDS Kurallarını Anlamak

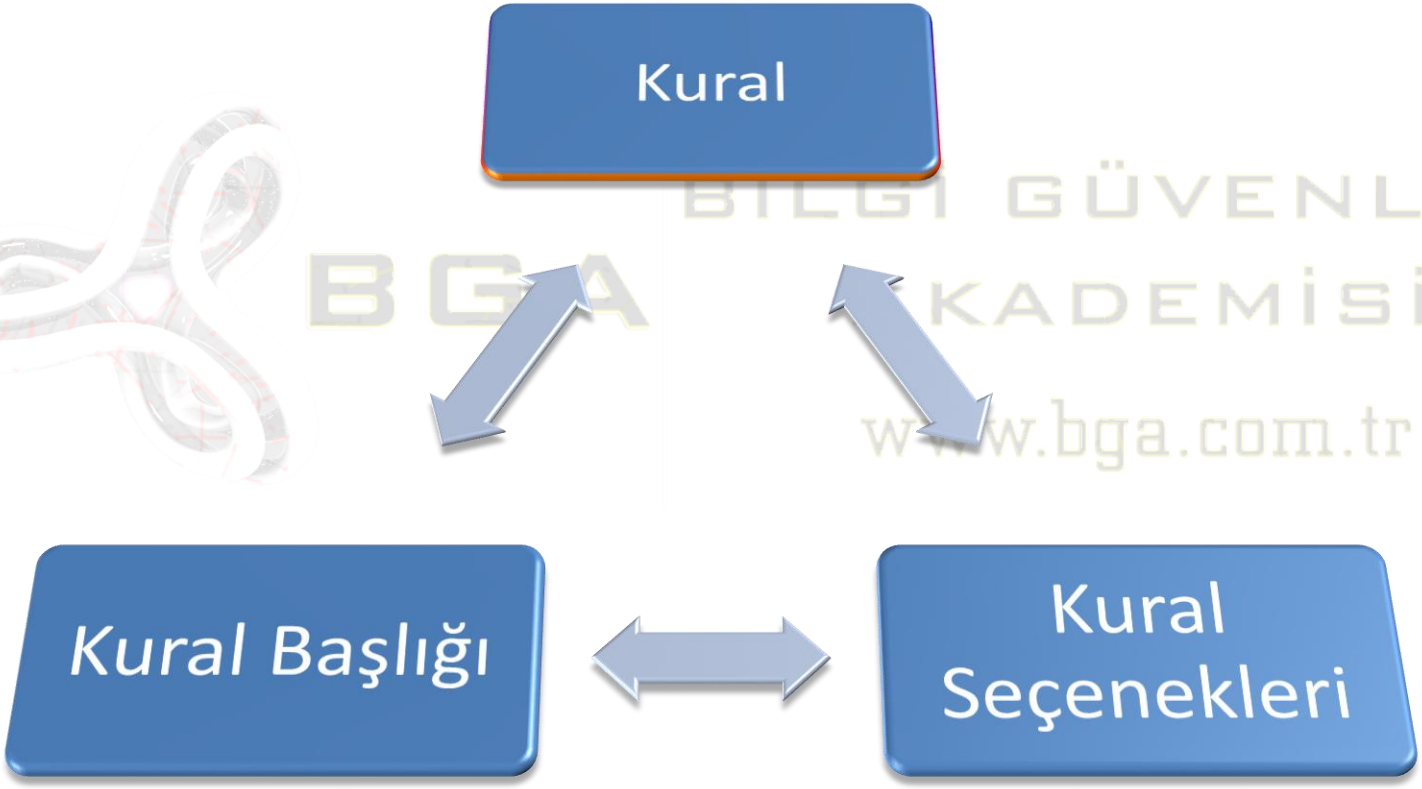
- Oldukça Esnek kural yazma imkanı
- Hazır kuralları kullanma
  - BleedingEdge
  - SourceFire Kuralları
  - Kuralları Güncelleme -OinkMaster
- Kural = Kural Başlığı + Kural Seçenekleri

Basit IDS Kuralı




```
alert tcp $TELNET_SERVERS 23 -> $EXTERNAL_NET any (msg:"TELNET root
    login"; content:"login\: root";
flow:from_server,established; classtype:suspicious-login; sid:719; rev:5;)
```

Snort Kuralları



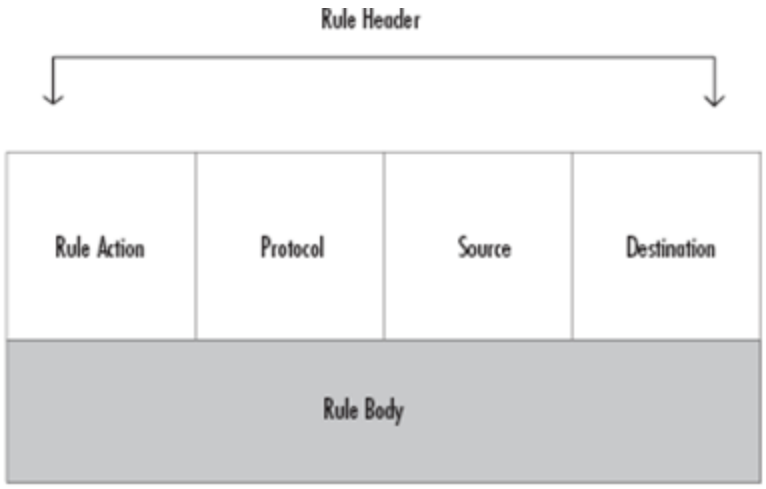
Kural Başlık/Seçenekleri

- Her kuralda bir adet kural başlığı ve kural seçeneği bulunur.
- Snort Kurallarının gücü kural seçeneklerindedir.
- Kural başlıkları Firewall benzeri mantıkla çalışır.

Kural Başlığı	Kural Seçenekleri
Alert tcp 1.1.1.1 any -> 2.2.2.2 any	(flags: SF; msg: “SYN-FIN Scan”;)
Alert tcp 1.1.1.1 any -> 2.2.2.2 any	(flags: S12; msg: “Queso Scan”;)
Alert tcp 1.1.1.1 any -> 2.2.2.2 any	(flags: F; msg: “FIN Scan”;)

Kural Başlığı

- **alert tcp ! $EXTERNAL_NET any -> $TELNET_SERVERS 23**
- **Kural başlığı:** paketin nerden gelip nereye gittiğine , çeşidine(tcp, udp, icmp, ip vs) ve kurala uyan paketlerin akibetine karar verir.
- Alert/log/pass/activate/dynamic/drop/sdrop/reject.
- Tek bir IP adresi, CIDR, gruplama kullanılabilir.
- Kural başlığı 4 alt bölüme ayrılır:
  - Kural Aksiyonu
  - Protokol
  - Kaynak (IP Port)
  - Hedef (IP Port)



Kural Başlığı:Aksiyon

- Snort kural başlığının en önemli alanlarından biridir ve imzaya uyan paket için ne yapılacağını belirtir.

```
alert tcp 192.168.0.0/24 any -> 10.10.10.180
```

Aksiyon	İşlevi
Alert	Uyan paketler için uyarı vermek ve loglamak için
Log	Uyarı vermden sadece loglamak için
Pass	Paketi önemseme
Activate	Uyarı verip dinamik bir kuralı tetiklemek için
Dynamic	Activate aracılığı ile gelen emirleri bekleyerek işleme almak için
Drop	Iptables'ın paketi bloklaması ve loglaması için
Sdrop	Iptables'in paketi bloklaması için.(Loglama yok)
Reject	Iptables'in saldırgana TCP RST ya d aIcmp port unreachabel mesajı göndererek loglaması için.

Kural Başlığı: Protokol Alanı

alert tcp 192.168.0.0/24 any -> 10.10.10.180

Hangi Protokolü incelediğini belirtir
Aşağıdaki değerleri alabilir
- **TCP**
- **UDP**
- **ICMP**
- **IP**

Kural Başlığı: IP Adres Alanı

```
alert tcp 192.168.0.0/24 any -> 10.10.10.180
```

Kaynak IP: Trafiğin nerden geldiğini belirtir
- CIDR olabilir
- Tek bir IP Adresi olabilir
- Netmask olabilir
- Önüne ! Koyarak hariç tutulabilir
- Any özel kelimesiyle tüm IP adresleri kastedilebilir
- $HOME_NET gibi değişken tanımları kullanılabilir

Kural Başlığı: IP Adres Alanı

alert tcp 192.168.0.0/24 any -> 10.10.10.180

HedefIP: Trafiğin nerye gittiğini belirtir
- CIDR olabilir
- Tek bir IP Adresi olabilir
- Netmask olabilir
- Önüne ! Koyarak hariç tutulabilir
- Any özel kelimesiyle tüm IP adresleri kastedilebilir
- $HOME_NET gibi değişken tanımları kullanılabilir

Kural Başlığı: Port Alanı

alert tcp 192.168.0.0/24 any -> 10.10.10.1 80

Port : Trafiğin hangi porttan gelip hangi porta gittiğini belirtir

- 80, 110, 443 gibi bir değer alabilir
- Önüne ! Koyarak hariç tutulabilir (!80)
- Any özel kelimesiyle tüm port numaraları kapsanabilir
- 22:900 gibi aralık verilebilir
- $ORACLE_PORTS gibi değişken tanımları kullanılabilir
- Büyüktür, küçüktür ifadeleri kullanılabilir (:1024, 2200:) gibi

Kural Başlığı:Yön

alert tcp 192.168.0.0/24 any -> 10.10.10.180

Trafiğin sol taraftan sağ tarafa doğru aktığını belirtir
->
<>
İfadeleri kullanılabilir

Kural Seçenekleri

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS \
(msg:"WEB-MISC rcmd attempt"; flow:to_server,established; uricontent:"rcmd.exe"; nocase; \
metadata:service http; classtype:web-application-activity; sid:1065; rev:8;)
```

Kural Başlığı

Kural Seçenekleri

- Detection Engine'nin kalbi sayılır
- () arasına yazılır ve birbirinden ";" ile ayrılır
- () arasına almak zorunludur
- Her seçenek ; ile biter, son seçenek dahil!
- Seçenek ve alacağı değer : ile ayrılır
- Meta-data, payload, non-payload, post-detection alanlarına ayrılır

Metadata

- Kural hakkında çeşitli bilgiler vermek için
  - Raportlama ve analiz aşamasında kullanılır
- Msg Kural tetiklendiğinde verilecek mesaj
  - (msg:"WEB-MISC rcmd attempt"; ...
- Sid Snort kural ID
  - sid:1065;
  - Kural numaraları(0-100 arası kullanılmaz)
- Rev Kuralın kaç revizyon geçirdiği
  - id:1065; rev:8;

Metadata-II

- Reference: Tetiklenen kuralla ilgili detay bilgileri içeren referanslar
  - reference:url,www.securiteam.com/exploits/3D5Q4RFPPK.html;
- Classification: Kuralı sınıflandırma amaçlı
  - classtype: trojan-activity;
- Priority: Kurala önem tanımlama
  - Düşük değer daha yüksek öneme sahiptir

Kural Yazımı-Non Payload Detection

- Protokollerin başlıkları ile ilgilenir
  - TTL Alanı kontrolü ttl:<3;
- IP Tos Alanı kontrolü tos:8; (Minimize Delay)
- Ipopts Alanı Kontrolu
  - Record route, IP security option , Loose source routing , any IP options are set
- Fragbits
  - IP parçalanma alanını kontrol eder
- Flags: TCP Bayraklarını kontrol eder
  - (msg:"SCAN nmap XMAS"; stateless; flags:FPU,12;

http://www.procyonlabs.com/snort_manual/node1.html

Kural Seçenekleri:IP

- Fragoffset
- Ttl
- id
- Tos
- ipopts
- Fragbits
- Dsize
- ip_proto
- Sameip

Kural Seçenekleri:IP->TTL

- IP başlığındaki TTL alanını kontrol etmek için kullanılır
  - ttl:[[<number>-]><=]<number>;
- Örnek kullanım
  - ttl:<2;
  - ttl:1-3;
- Traceroute yakalama

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP traceroute"; itype:8; ttl:<2; reference:arachnids,118; classtype:attempted-recon; sid:385; rev:4;)

Kural Seçenekleri:IP->sameip

- Kaynek ve hedef ip adreslerinin aynı olup olmadığını kontrol eder
  - Land attack
- alert ip any any -> any any (sameip;)

Kural Seçenekleri:IP->ipopts

- Gelen-giden paketlerde herhangi bir IP seçeneğinin olup olmadığını kontrol eder

```
alert ip $EXTERNAL_NET any -> $HOME_NET any
(msg:"MISC source route ssrr"; ipopts:ssrr ;
reference:cve,1999-0510; classtype:bad-unknown;
sid:502; rev:4;)
```

rr
- Record Route

eol
- End of list

nop
- No Op

ts
- Time Stamp

sec
- IP Security

esec
- IP Extended Security

lsrr
- Loose Source Routing

ssrr
- Strict Source Routing

satid
- Stream identifier

any
- any IP options are set

Kural Seçenekleri:TCP

- TCP başlık bilgilerini kontrol etmek için kullanılır.
  - Flags
  - Seq
  - Ack
  - Flow
  - stateless

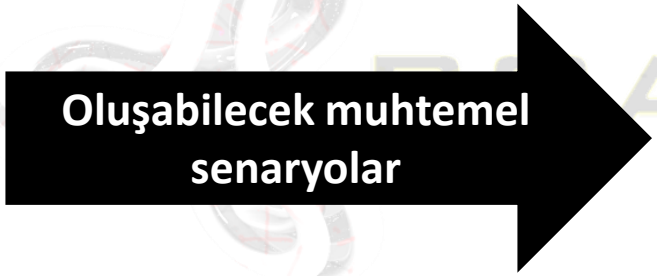
Kural Seçenekleri:TCP->Flags

- Gelen pakette hangi bayrakların set edildiğini bulmaya yarar
  - flags:[!|*|+]<FSRPAU120>[,<FSRPAU120>];


```
alert tcp any any -> any any (flags:SF,12;)
```

Kural Seçenekleri:TCP->Flow

- TCP oturum durumunu kontrol eder



Option	Description
to_client	Trigger on server responses from A to B
to_server	Trigger on client requests from A to B
from_client	Trigger on client requests from A to B
from_server	Trigger on server responses from A to B
established	Trigger only on established TCP connections
stateless	Trigger regardless of the state of the stream processor (useful for packets that are designed to cause machines to crash)
no_stream	Do not trigger on rebuilt stream packets (useful for dsize and stream5)
only_stream	Only trigger on rebuilt stream packets

alert tcp !$HOME_NET any -> $HOME_NET 21 (msg:"cd incoming detected"; \ flow:from_client; content:"CWD incoming"; nocase;)

Kural Seçenekleri:ICMP

- ICMP başlık bilgilerini inceleyen kural seçeneği
- Itype
- Icode
- Icmp_seq
- Icmp_id

Değerlerini alabilir.

Kural Seçenekleri: Dsize

- Paket “payload” kısmının boyutunu ölçmek için kullanılır.
  - dsize: [<>]<number>[<><number>];

alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 1433:1500 (msg:"SQL Microsoft SQL Server 2000 Server hello buffer overflow attempt"; flow:to_server,established; **dsize:>511;** content:"|12 01|"; depth:2; content:!"|00|"; within:512; distance:35; reference:bugtraq,5411; reference:cve,2002-1123; reference:url,www.microsoft.com/technet/security/Bulletin/MS02-056.mspx; classtype:attempted-admin; sid:11264; rev:5;)

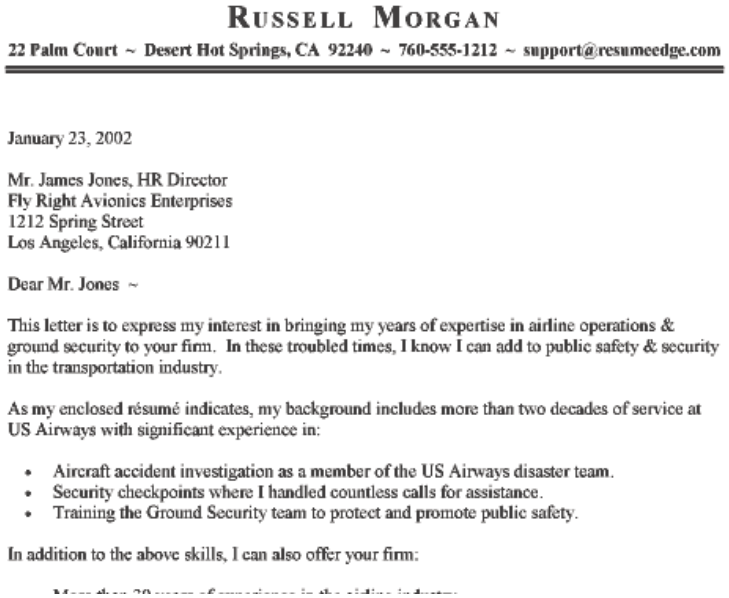
Tüm Non-Payload Seçenekleri

Keyword	Description
fragoffset	The fragoffset keyword allows one to compare the IP fragment offset field against a decimal value.
ttl	The ttl keyword is used to check the IP time-to-live value.
tos	The tos keyword is used to check the IP TOS field for a specific value.
id	The id keyword is used to check the IP ID field for a specific value.
ipopts	The ipopts keyword is used to check if a specific IP option is present.
fragbits	The fragbits keyword is used to check if fragmentation and reserved bits are set in the IP header.
dsize	The dsize keyword is used to test the packet payload size.
flags	The flags keyword is used to check if specific TCP flag bits are present.
flow	The flow keyword allows rules to only apply to certain directions of the traffic flow.
flowbits	The flowbits keyword allows rules to track states during a transport protocol session.

seq	The seq keyword is used to check for a specific TCP sequence number.
ack	The ack keyword is used to check for a specific TCP acknowledge number.
window	The window keyword is used to check for a specific TCP window size.
itype	The itype keyword is used to check for a specific ICMP type value.
icode	The icode keyword is used to check for a specific ICMP code value.
icmp_id	The icmp_id keyword is used to check for a specific ICMP ID value.
icmp_seq	The icmp_seq keyword is used to check for a specific ICMP sequence value.
rpc	The rpc keyword is used to check for a RPC application, version, and procedure numbers in SUNRPC CALL requests.
ip_proto	The ip_proto keyword allows checks against the IP protocol header.
sameip	The sameip keyword allows rules to check if the source ip is the same as the destination IP.

Payload

- Paketin içerisini incelenen kural seçenek kısmı



RUSSELL MORGAN

22 Palm Court ~ Desert Hot Springs, CA 92240 ~ 760-555-1212 ~ support@resumeedge.com

January 23, 2002

Mr. James Jones, HR Director
Fly Right Avionics Enterprises
1212 Spring Street
Los Angeles, California 90211

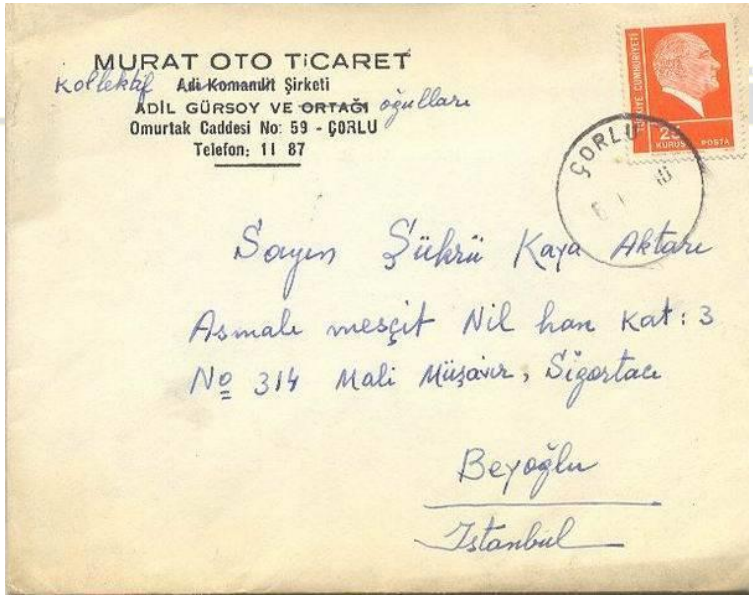
Dear Mr. Jones ~

This letter is to express my interest in bringing my years of expertise in airline operations & ground security to your firm. In these troubled times, I know I can add to public safety & security in the transportation industry.

As my enclosed résumé indicates, my background includes more than two decades of service at US Airways with significant experience in:

- Aircraft accident investigation as a member of the US Airways disaster team.
- Security checkpoints where I handled countless calls for assistance.
- Training the Ground Security team to protect and promote public safety.

In addition to the above skills, I can also offer your firm:



MURAT OTO TİCARET
Kollektif Adi Komandit Şirketi
ADİL GÜRSOY VE ORTAĞI oğulları
Omurtak Caddesi No: 59 - ÇORLU
Telefon: 11 87

ÇORLU

Sayın Şükrü Kaya Aktaru
Asmalı mescit Nil han Kat: 3
No 314 Mali Müşavir, Sigortacı
Beyoğlu
İstanbul

PAYLOAD

HEADER

Kural Seçenekleri:Content

- Paket veri alanında spesifik içerik tarama için kullanılır.
  - content: [!] "<content string>";
- Binary(ikili) içerik için | 00 0F| kullanılır (hex)
- Bir kural da hem text hem hex değerler bulunabilir.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 143 (msg:"IMAP login brute force attempt"; flow:to_server,established; content:"LOGIN"; nocase;
```

Kural Seçenekleri:Nocase

- Content için arama yapılırken büyük küçük harf ayrımı yapılmayacağını belirtir.


```
alert tcp any any -> any 21 (msg:"FTP ROOT";
content:"USER root"; nocase;)
```

Kural Seçenekleri:Offset

- "Content" için arama işleminin payload'un neresinden başlanacağını belirtir
  - 300K lık bir paket içerisinde 3K'lık bir arama için tüm paketi dolaşmak gereksiz ve perfomans yorucudur
- Offset bir önceki content: tanımını etkiler

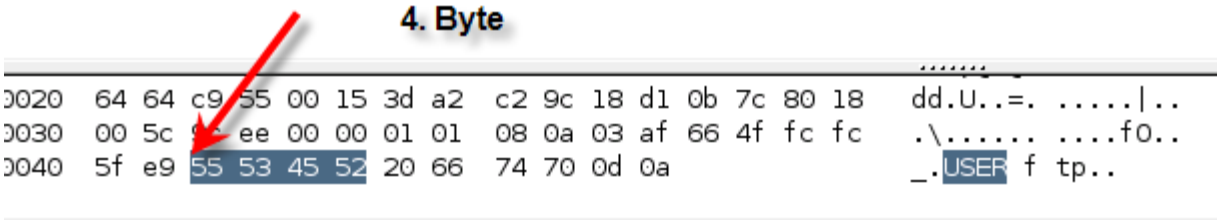

```
alert tcp any any -> any 80 (content: "cgi-bin/phf";
offset:4; depth:20;)
```

Kural Seçenekleri:Depth

- Snort'un kaç byte'lık dilime bakacağını belirtir
- Bir önceki "content:" seçeneğini etkiler

```
alert tcp any any -> any 80 (content: "cgi-bin/phf"; offset:4;
depth:20;)
```

- 4. Byte'dan başla 20 Byte incele...



Kural Seçenekleri:Distance

- Bir önceki “content:” tanımlamasından ne kadar byte ileri gidileceğini belirtir
- Content:”A”; content:”C”; distance:1
  - A ile C arasında bir boşluk var
- ABC ile başlayıp -arada bir karekter herhangi birşey gelebilir- DEF ile biten içerik araması

```
alert tcp any any -> any any (content:"ABC"; content:
"DEF"; distance:1;)
```

Kural Seçenekleri:Within

- Bir önceki “content:” den sonra ne kadarlık bir alan içerisinde ikinci “content” in araştırılacağını belirler.
- ABC’den sonra 10 byte içerisinde EFG ara

```
alert tcp any any -> any any (content:"ABC"; content: "EFG"; within:10;)
```

Kural Seçenekleri:UriContent

- HTTPInspect önişlemcisi tarafından normalleştirilmiş HTTP trafiği içerisindeki URL kısmını inceler
- uricontent:[!]<content string>;

alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"SPYWARE-PUT Adware gophoria toolbar runtime detection"; flow:to_server,established; **uricontent:"/application/app_counter/?gopver=";** nocase; reference:url,www.360zd.com/spyware/518.html; classtype:misc-activity; sid:12791; rev:1;)

Görünmez İçerik Filtreleme Sistemi

- URIContent özelliği kullanılarak L2 modda (üzerinde IP adresi olmayan) içerik filtreleme yapılabilir.
  - Uricontent:”http://www.google.com”
- Youtube yasağına çözüm!
  - Uricontent:”http://www.youtube.com/videoid?901”

Kural Seçenekleri:http_header

- HTTP istek ve cevapları için kullanılır.

```
alert http $HOME_NET any -> $EXTERNAL_NET any (\
msg:"ET P2P ABC Torrent User-Agent (ABC/ABC-3.1.0)"; \
header.useragent:"ABC/ABC"; \
sid:2003475;)
```

Kural Seçenekleri:Pcre

- PCRE = Perl compatible regular expressions
- Regex yazım kuralları bilinmelidir
  - http://www.pcre.org
- Performans canavarıdır!
  - Çok gerekmedikçe kullanılmamalıdır.


```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-PHP gallery
arbitrary command execution attempt"; flow:to_server,established; uricontent:"/setup/";
content:"GALLERY_BASEDIR="; pcre:"/GALLERY_BASEDIR=(http|https|ftp)/i";
reference:nessus,11876; reference:bugtraq,8814; classtype:web-application-attack; sid:2306;
rev:2;)
```

Tüm Payload Kural Seçenekleri

content	The content keyword allows the user to set rules that search for specific content in the packet payload and trigger response based on that data.
rawbytes	The rawbytes keyword allows rules to look at the raw packet data, ignoring any decoding that was done by preprocessors.
depth	The depth keyword allows the rule writer to specify how far into a packet Snort should search for the specified pattern.
offset	The offset keyword allows the rule writer to specify where to start searching for a pattern within a packet.
distance	The distance keyword allows the rule writer to specify how far into a packet Snort should ignore before starting to search for the specified pattern relative to the end of the previous pattern match.
within	The within keyword is a content modifier that makes sure that at most N bytes are between pattern matches using the content keyword.
uricontent	The uricontent keyword in the Snort rule language searches the normalized request URI field.

isdataat	The isdataat keyword verifies that the payload has data at a specified location.
pcre	The pcre keyword allows rules to be written using perl compatible regular expressions.
byte_test	The byte_test keyword tests a byte field against a specific value (with operator).
byte_jump	The byte_jump keyword allows rules to read the length of a portion of data, then skip that far forward in the packet.
ftpbounce	The ftpbounce keyword detects FTP bounce attacks.
asn1	The asn1 detection plugin decodes a packet or a portion of a packet, and looks for various malicious encodings.
cvs	The cvs keyword detects invalid entry strings.
dce_iface	See the DCE/RPC 2 Preprocessor section.
dce_opnum	See the DCE/RPC 2 Preprocessor section.
dce_stub_data	See the DCE/RPC 2 Preprocessor section.

Post-detection(Kural Aksiyonu Belirleme)

- Kuralın ne aksiyon alacağını belirler
  - Logto
  - Session
  - Resp
  - React
  - Tag
  - Replace
  - Detection_filter

Gibi alt alanlardan oluşur.

Kural Aksiyonu: logto

- Kuralın tetikleyen trafiği harici bir dosyaya kaydetmek için kullanılır.
  - logto:"filename";

Kural Aksiyonu:Session

- TCP oturumlarından veri ayıklama amaçlı kullanılır.
- session: [printable|all];
- log tcp any any <> any 23 (session:printable;)
- Telnet oturumlarındaki okunabilir trafiği kaydeder.
- All = doğrudan okunabilir olmayan(ornek: binary dosyalar) kaydetme için

Kural Aksiyonu:Resp

- TCP/UDP bağlantılarını sonlandırma amaçlı kullanılır
  - Flexresp özelliği derlemede eklenmiş olmalı
- alert tcp any any -> any 80 (resp:rst_all;)
  - 80.porta giden tüm isteklere RST gönder

Option	Description
rst_snd	Send TCP-RST packets to the sending socket
rst_rcv	Send TCP-RST packets to the receiving socket
rst_all	Send TCP_RST packets in both directions
icmp_net	Send a ICMP_NET_UNREACH to the sender
icmp_host	Send a ICMP_HOST_UNREACH to the sender
icmp_port	Send a ICMP_PORT_UNREACH to the sender
icmp_all	Send all above ICMP packets to the sender

Kural Aksiyonu:React

- Snort'un içerik filtreleme amaçlı kullanılmasını sağlar
- Kullanıcı bir siteye erişmek istediğine
  - Engelleyebilir
  - Engelleyip uyarı çıkarabilir
  - Engelleyip başka bir adrese(websense vs gibi) yönlendirebilir

```
alert tcp any any <> 192.168.1.0/24 80 (content: "bad.htm"; \ msg: "Not for children!"; react: block, msg, proxy 8000;)
```

Kural Aksiyonu:detection_filter

- Bir kuralın event üretmesini bir eşik değerine bağlar
- detection_filter: \ track <by_src|by_dst>, \ count <c>, seconds <s>;
- 10.10.10.1 IP adresine 60 saniye içerisinde 30 adet SSH isteği gelirse engelle!

```
drop tcp any any > 10.10.10.1 22 ( \ msg:"SSH Brute Force Attempt";
flow:established,to_server; \ content:"SSH"; nocase; offset:0; depth:4; \
detection_filter: track by_src, count 30, seconds 60; \ sid:1000001; rev:1;)
```

Kural Sıralaması

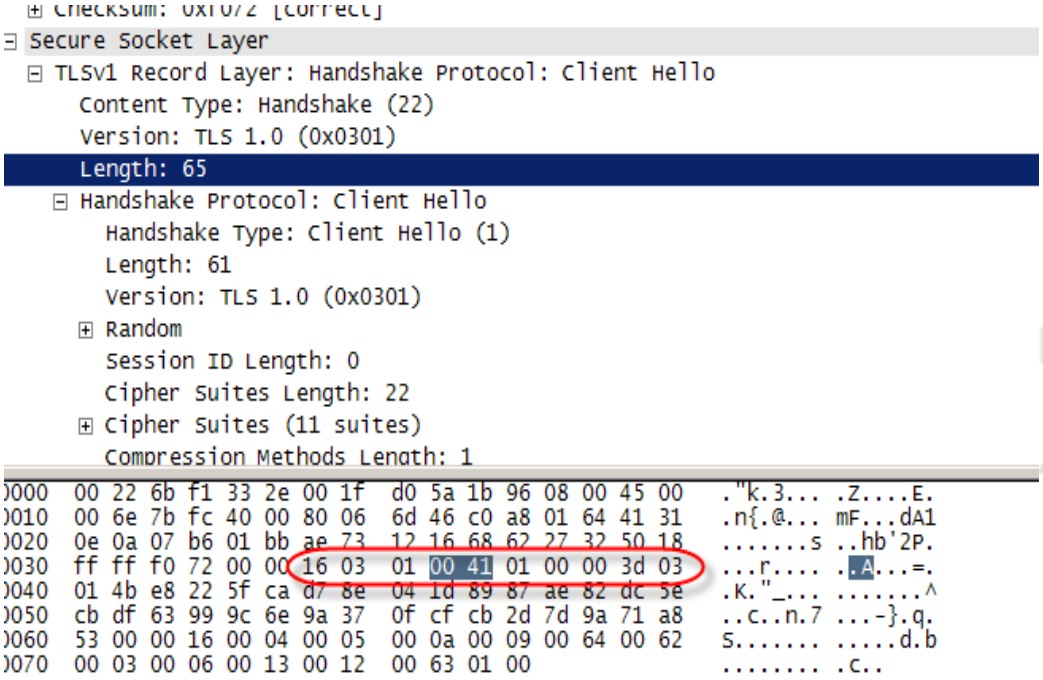
- Kural aksiyonlarında yer alan Alert, pass, log gibi ifadelerin hangisinin öncelikli olduğunu belirler.
- Öntanımlı değer:alert->pass->log
- Snort.conf'da "config order:" veya komut satırından –o parametresi

```
root@seclabs:/etc/snort# man snort
       -o      Change the order in which the rules are applied to packets.  Instead of being applied
               in the standard Alert->Pass->Log order, this  will  apply  them  in  Pass->Alert->Log
               order.
```

Kural Sıralaması-II

- Nerde işe yarar?
- Bazı kuralların belirli IP adresleri için uyarı vermemesi istenilebilir
  - Vulnerability Scanner cihazının tüm trafiği IPS tarafından izlenmekte ve her taramada uyarı vermekte!
  - Pass kuralı yazılarak Vuln.Scan cihazının trafiğinin IDS tarafından loglanması engellenir.
  - Dogrudan bpf yazılarak Snort'un belirli ip adreslerinden gelen trafiğe hiç dokunmaması sağlanabilir.
    - Snort not src host 10.10.10.1 gibi...

Ultrasurf Engelleme Kuralı



1603010041010000 3d0301 hex ifadesinde normal TLS bağlantılarından farklı tek şey Length değerleri.

16: Content Type: Handshake

03 01: Version TLS1.0

00 41: Length 65

01: Handshake Type: Client Hello

00 00 3d: Length 61

03 01:Version TLS1.0

alert tcp $HOME_NET any -> $EXTERNAL_NET 443 (msg:"Ultrasurf Kullanimi!"; flow:to_server,established; content:"|1603010041010000 3d0301|"; classtype:policy-violation; sid:1000099;)

Hangi Kural Ne İşe Yarar?

Sid	Summary	Impact	Detailed Information	Affected Systems	Attack Scenarios	Ease of Attack	False Positive	False Negative	Corrective Action	Contributors	Additional References
1:1520	This event is generated when an attempt is made to access server-info. Using the Apache webserver, this url is generally handled by the mod_info module, which will happily disclose valuable information about your webserver which may aid in their attack.	Information disclosure.	The mod_info module "provides a comprehensive overview of the server configuration including all installed modules and directives in the configuration files" for the Apache webserver. Successfully accessing the url that is handle by mod_info may give an attacker valuable information about the server. If mod_info is in use and the attacking host is allowed to access it, every possible configuration option that the Apache server is using can be viewed. This includes ACLs, modules, file and directory names, and other valuable information that will help an attacker determine ways of attacking the server.	Apache webservers with mod_info enabled.	As part of an attack against an Apache webserver, an attacker may try to access "/server-info" which is typically handled by the mod_info module. If sucessful, this will give valuable information about the webserver for use in further attacks.	Simple. No exploit software is required.	Few, but certainly possible. Since this rule only checks for the existance of "/server-info" in the url, any url containing that string will trigger this rule. A few common false positives may include urls like: http://victim /server-info/contact.html http://victim /really /long/directory /server-info.html	None Known	Determine if server-info exists on the victim in question, and if the attacker is allowed to access it. If mod_info is necessary on this server, consider restricting access to it via Apache directives, i.e.: SetHandler server-info Order deny,allow Deny from all Allow from .yourdomain.net	Snort documentation contributed by Jon Hart Sourcefire Vulnerability Research Team Brian Caswell Nigel Houghton	

Örnek Kural-#1

```
alert tcp $EXTERNAL_NET any <> $HOME_NET 0 (msg:"BAD-TRAFFIC tcp port 0 traffic";
flow:stateless; classtype:misc-activity; sid:524; rev:9;)
```

Örnek Kural-#2

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS zone transfer TCP";
flow:to_server,established; content:"|00 00 FC|"; offset:15; metadata:policy
security-ips drop, service dns; reference:arachnids,212; reference:cve,1999-0532;
reference:nessus,10595; classtype:attempted-recon; sid:255; rev:16;)
```

Örnek Kural-#3

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP traceroute";
itype:8; ttl:1; reference:arachnids,118; classtype:attempted-recon; sid:385; rev:4;)
```

Örnek Kural-#4

- Port tarama(nmap -sS)

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"ET SCAN NMAP -
sS window 2048"; fragbits:!D; dsize:0; flags:S,12; ack:0; window:20
48; classtype:attempted-recon;
reference:url,doc.emergingthreats.net/2000537;
reference:url,www.emergingthreats.net/cgi-bin/cvsweb.c
gi/sigs/SCAN/SCAN_NMAP; sid:2000537; rev:7;)
```

```
[**] [1:2009584:1] ET SCAN NMAP -sS window 4096 [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/29-23:24:06.440226 88.240.10.125:50548 -> 212.98.228.246:80
TCP TTL:52 TOS:0x0 ID:38946 IpLen:20 DgmLen:60
******S* Seq: 0x22700D04  Ack: 0x0  Win: 0x1000  TcpLen: 40
TCP Options (6) => MSS: 1452 NOP WS: 0 NOP NOP TS: 17745 0
[Xref => http://www.emergingthreats.net/cgi-bin/cvsweb.cgi/sigs/S
```

Örnek Kural-#5

- /etc/passwd

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC
/etc/passwd"; flow:to_server,established; content:"/etc/passwd"; nocase;
metadata:service http; classtype:attempted-recon; sid:1122; rev:6;)
```

Örnek Kural-#6

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"P2P Skype client login";
flow:to_client,established; flowbits:isset,skype.login; dsize:5; content:"|17 03 01 00|";
depth:4; metadata:policy security-ips drop; classtype:policy-violation; sid:5999; rev:4;)
```

Örnek Kural-#7


```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"P2P Skype client login";
flow:to_client,established; flowbits:isset,skype.login; dsize:5; content:"|17 03 01 00|";
depth:4; metadata:policy security-ips drop; classtype:policy-violation; sid:5999; rev:4;)
```

Örnek Kural-#8

- Syn flood yakalama/botnet detection

```
ddos.rules:# alert tcp $HOME_NET any <> $EXTERNAL_NET any (msg:"DDOS
shaft synflood"; flow:stateless; flags:S,12; seq:674711609; reference:cve,2000-
0138; classtype:attempted-dos; sid:241; rev:13;)
```

DNS DOS Saldırısı

```
alert udp $HOME_NET 53 -> $EXTERNAL_NET any (msg:"DOS DNS root
query response traffic amplification attempt";\
flow:to_client; content:"|00 01|"; depth:2; offset:4; content:"|00 00 02
00 01|"; within:5; distance:6;\
threshold:type threshold, track by_dst, count 5, seconds 30;
metadata:service dns;\
reference:url,isc.sans.org/diary.html?storyid=5713; classtype:misc-
activity; sid:15260; rev:1;)
```

Kuralları Güncelleme

- Kural güncelleme yöntemleri
  - Elle(her hafta yeni kurallar indirilirilerek)
  - Oinkmaster ile otomatik
- oinkmaster.pl -o /etc/snort/rules
  - Oinkmaster kodu gerektirir(snort.org'dan)

http://oinkmaster.sourceforge.net/readme.shtml

BGA İletişim

www.bga.com.tr

blog.bga.com.tr

twitter.com/bgasecurity

facebook.com/BGAkademisi

bilgi@bga.com.tr

egitim@bga.com.tr