

Mini Tehdit İstihbaratı

written by Mert SARICA | 3 December 2018

Siber Güvenlik Merkezleri tarafından kurumlara gerçekleştirilen hedefli siber saldırıları tespit etmenin ve engellemenin öneminin yüksek olduğu günümüzde, basit bir kontrol ile tespit edilebilen düşük etkileşimli bir balküpü sistemi (Bkz: Balküpü Tespiti) maalesef kurumlar için kaynakları tüketen atıl bir sistem olmaktan öteye gidemiyor. Halbuki kolay bir şekilde kurup, yönetebileceğimiz, amacına hizmet edebilen bir balküpü sistemi yeri geldiğinde SIEM ile entegre edilerek kurumumuz için oldukça değerli bir mini siber tehdit istihbarat servisi olarak da kullanılabilir.

Bu ev yapımı, mini istihbarat servisi ile ayrıca kurumların hangi basit şifreler ile hangi servisler, protokoller üzerinden hacklendiğini, hangi ülkelerin siber saldırganlara ev sahipliği yaptığını bulmak da mümkün olabilirdi. Tuzak Sistem ile Hacker Avı çalışmamda olduğu gibi evimde konumlandıracağım bu sistemin hem basit bir şekilde siber saldırganlar tarafından tespit edilememesi hem de kısıtlı zamanımı çalmaması adına kolay yönetilebilmesi gereksinimlerimin başında geliyordu.

Bu konu üzerine yeteri kadar düşündükten ve taşındıktan sonra açık kaynak kodlu çeşitli servisleri modifiye edip bu servise yapılan hatalı giriş denemelerini (failed login attempt) istediğim şekilde kayıt altına alan bir sistem oluşturmaya karar verdim. Hangi servisler olacağına karar vermek için ise güvenlik uzmanları dışında bir o kadar da art niyetli kişiler tarafından da sıklıkla kullanılan ncrack ve THC-Hydra araçlarının destekledikleri protokollere göz atmaya karar verdim ve günün sonunda ssh, ftp, http, postgresql protokollerinde karar kıldım. Daha sonra Mini-PC üzerine kurduğum XenServer sanallaştırma sistemi üzerine iki tane sanal Ubuntu işletim sistemi kurdum. Ubuntulardan birine bu denemeleri kayıt altına alan loginmon isimli aracı bir diğerine loginmon aracınının kayıtlarını (log) analiz etmek amacıyla Splunk Community sürümünü kurdum.


```

GNU nano 2.5.3 File: auth-passwd.c
#include "canohost.h"
extern Buffer loginmsg;
extern Serveroptions options;

#ifdef HAVE_LOGIN_CAP
extern login_cap_t *lc;
#endif

#define DAY (24L * 60 * 60) /* 1 day in seconds */
#define TWO_WEEKS (2L * 7 * DAY) /* 2 weeks in seconds */

void
disable_forwarding(void)
{
    no_port_forwarding_flag = 1;
    no_agent_forwarding_flag = 1;
    no_x11_forwarding_flag = 1;
}

/*
 * Tries to authenticate the user using password. Returns true if
 * authentication succeeds.
 */
int
auth_password(Authctxt *authctxt, const char *password)
{
    struct passwd *pw = authctxt->pw;
    int result, ok = authctxt->valid;
    #if defined(USE_SHADOW) && defined(HAS_SHADOW_EXPIRE)
    static int expire_checked = 0;
    #endif

    /* MERT SARICA */
    logit("[Failed Password] Client: %s, Username: %s Password: %s", get_remote_ipaddr(), authctxt->user, password);

#ifdef HAVE_CYGWIN
    if (pw->pw_uid == 0 && options.permit_root_login != PERMIT_YES)
        ok = 0;
#endif

    if (*password == '\0' && options.permit_empty_passwd == 0)
        return 0;

#ifdef KRB5
    if (options.kerberos_authentication == 1) {
        int ret = auth_krb5_password(authctxt, password);
        if (ret == 1 || ret == 0)
            return ret && ok;
        /* Fall back to ordinary passwd authentication. */
    }
#endif

#ifdef HAVE_CYGWIN
    HANDLE hToken = cygrin_logon_user(pw, password);
    if (hToken == INVALID_HANDLE_VALUE)
        return 0;
    cygrin_set_impersonation_token(hToken);
    return ok;
#endif

#ifdef USE_PAM
    if (options.use_pam)
        return (ssh_pam_auth_passwd(authctxt, password) && ok);
#endif

    #if defined(USE_SHADOW) && defined(HAS_SHADOW_EXPIRE)
    if (!expire_checked) {

```

```

root@kali:~# cat pass.txt
test
mert
dert
root@kali:~# hydra -t 32 -l root -P pass.txt 192.168.1.127 ssh
Hydra v8.3 (C) 2016 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-11-07 11:06:04
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 3 tasks per 1 server, overall 64 tasks, 3 login tries (1:1/p:3), ~0 tries per task
[DATA] attacking service ssh on port 22
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-11-07 11:06:05
root@kali:~#

```

```

root@ubuntu:~# tail -n 9 /var/log/auth.log
Nov 7 19:06:05 ubuntu sshd[20604]: [Failed Password] Client: 192.168.1.144 Username: root Password: mert
Nov 7 19:06:05 ubuntu sshd[20603]: [Failed Password] Client: 192.168.1.144 Username: root Password: dert
Nov 7 19:06:05 ubuntu sshd[20602]: [Failed Password] Client: 192.168.1.144 Username: root Password: test
Nov 7 19:06:05 ubuntu sshd[20604]: Failed password for root from 192.168.1.144 port 64781 ssh2
Nov 7 19:06:05 ubuntu sshd[20603]: Failed password for root from 192.168.1.144 port 64780 ssh2
Nov 7 19:06:05 ubuntu sshd[20602]: Failed password for root from 192.168.1.144 port 64779 ssh2
Nov 7 19:06:05 ubuntu sshd[20604]: connection closed by 192.168.1.144 port 64781 [preauth]
Nov 7 19:06:05 ubuntu sshd[20603]: connection closed by 192.168.1.144 port 64780 [preauth]
Nov 7 19:06:05 ubuntu sshd[20602]: connection closed by 192.168.1.144 port 64779 [preauth]
root@ubuntu:~#

```

```

GNU nano 2.5.3 File: auth.c

/* Do not echo password to logs, for security. */
elog(DEBUG3, "received password packet");

/*
 * Return the received string. Note we do not attempt to do any
 * character-set conversion on it; since we don't yet know the client's
 * encoding, there wouldn't be much point.
 */
return buf.data;
}

/*-----
 * Password-based authentication mechanisms
 *-----*/

/* Plaintext password authentication.
 */
static int
checkPasswordAuth(port *port, char **logdetail)
{
    char *passwd;
    int result;
    char *shadow_pass;

    sendAuthRequest(port, AUTH_REQ_PASSWORD, NULL, 0);
    passwd = recv_password_packet(port);
    if (passwd == NULL)
        return STATUS_EOF; /* client wouldn't send password */

    /* Mert SARICA */
    elog(LOG, "[Failed Password] Username: %s Password: %s", port->user_name, passwd);
    shadow_pass = get_role_password(port->user_name, logdetail);
    if (shadow_pass)
        result = plain_crypt_verify(port->user_name, shadow_pass, passwd,
                                   logdetail);
    else
        result = STATUS_ERROR;

    if (shadow_pass)
        pfree(shadow_pass);
    pfree(passwd);
    return result;
}

/* MD5 and SCRAM authentication.
 */
static int
checkPwChallengeAuth(port *port, char **logdetail)
{
    int auth_result;
    char *shadow_pass;
    passwordType pwtype;

    Assert(port->hba->auth_method == uasCRAM ||
           port->hba->auth_method == uasMD5);

    /* First look up the user's password. */
    shadow_pass = get_role_password(port->user_name, logdetail);
}

```

```

Batcave x Batcave (1) Batcave (3) Batcave (2) Batcave (4)
root@ubuntu:~/honeypot# hydra -l root -P pass.txt 127.0.0.1 postgres
Hydra v8.6 (C) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-11-05 15:56:53
[DATA] max 3 tasks per 1 server, overall 3 tasks, 3 login tries (1:1/p:3), ~1 try per task
[DATA] attacking postgres://127.0.0.1:5432/
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-11-05 15:56:54
root@ubuntu:~/honeypot# cat /var/log/pgsql
2017-11-05 15:56:27.206 +03 [10974] LOG: listening on IPv6 address ":::1", port 5432
2017-11-05 15:56:27.206 +03 [10974] LOG: listening on IPv4 address "127.0.0.1", port 5432
2017-11-05 15:56:27.208 +03 [10974] LOG: listening on Unix socket "/tmp/.s.PGSQL.5432"
2017-11-05 15:56:27.228 +03 [10975] LOG: database system was shut down at 2017-11-05 15:54:28 +03
2017-11-05 15:56:27.231 +03 [10974] LOG: database system is ready to accept connections
2017-11-05 15:56:54.091 +03 [10991] LOG: [Failed Password] Username: root Password: dert
2017-11-05 15:56:54.091 +03 [10991] FATAL: password authentication failed for user "root"
2017-11-05 15:56:54.091 +03 [10991] DETAIL: Role "root" does not exist.
Connection matched pg_hba.conf line 86: "host all 0.0.0.0/0 password"
2017-11-05 15:56:54.094 +03 [10990] LOG: incomplete startup packet
2017-11-05 15:56:54.095 +03 [10992] LOG: [Failed Password] Username: root Password: mert
2017-11-05 15:56:54.095 +03 [10992] FATAL: password authentication failed for user "root"
2017-11-05 15:56:54.095 +03 [10992] DETAIL: Role "root" does not exist.
Connection matched pg_hba.conf line 86: "host all 0.0.0.0/0 password"
2017-11-05 15:56:54.096 +03 [10989] LOG: incomplete startup packet
2017-11-05 15:56:54.099 +03 [10994] LOG: [Failed Password] Username: root Password: test
2017-11-05 15:56:54.099 +03 [10994] FATAL: password authentication failed for user "root"
2017-11-05 15:56:54.099 +03 [10994] DETAIL: Role "root" does not exist.
Connection matched pg_hba.conf line 86: "host all 0.0.0.0/0 password"
2017-11-05 15:56:54.103 +03 [10993] LOG: incomplete startup packet
root@ubuntu:~/honeypot#

```

```
GNU nano 2.5.3 File: /root/honeypot/vsftpd-3.0.3/prelogin.c
if (tunable_userlist_enable)
{
    int located = str_contains_line(&p_sess->userlist_str, &p_sess->user_str);
    if ((located && tunable_userlist_deny) ||
        (!located && !tunable_userlist_deny))
    {
        check_login_delay();
        vsf_cmdio_write(p_sess, FTP_LOGINERR, "Permission denied.");
        check_login_fails(p_sess);
        str_empty(&p_sess->user_str);
        return;
    }
}
if (is_anon && tunable_no_anon_password)
{
    /* Fake a password */
    str_alloc_text(&p_sess->ftp_arg_str, "no password-");
    handle_pass_command(p_sess);
}
else
{
    vsf_cmdio_write(p_sess, FTP_GIVEPWD, "Please specify the password.");
}
}

static void
handle_pass_command(struct vsf_session* p_sess)
{
    if (str_isempty(&p_sess->user_str))
    {
        vsf_cmdio_write(p_sess, FTP_NEEDUSER, "Login with USER first.");
        return;
    }
}

/* Mert SARICA */
struct mystr str_log_line = INIT_MYSTR;
str_alloc_text(&str_log_line, "[Failed Password] client: ");
str_append_str(&str_log_line, &p_sess->remote_ip_str);
str_append_text(&str_log_line, " username: ");
str_append_str(&str_log_line, &p_sess->user_str);
str_append_text(&str_log_line, " password: ");
str_append_str(&str_log_line, &p_sess->ftp_arg_str);
vsf_log_line(p_sess, kvsfLogEntryConnection, &str_log_line);

/* These login calls never return if successful */
if (tunable_one_process_model)
{
    vsf_one_process_login(p_sess, &p_sess->ftp_arg_str);
}
else
{
    vsf_two_process_login(p_sess, &p_sess->ftp_arg_str);
}
vsf_cmdio_write(p_sess, FTP_LOGINERR, "Login incorrect.");
check_login_fails(p_sess);
str_empty(&p_sess->user_str);
/* FALLTHRU if login fails */
}

static void check_login_delay()
{
    if (tunable_delay_failed_login)
    {
        vsf_sysutil_sleep((double) tunable_delay_failed_login);
    }
}

static void check_login_fails(struct vsf_session* p_sess)
{
    str_append_text(p_str, "o ");
    /* Access mode: anonymous/real user, and identity */
    if (p_sess->is_anonymous && !p_sess->is_guest)
    {
        str_append_text(p_str, "a ");
        str_append_str(p_str, &p_sess->anon_pass_str);
    }
    else
    {
        if (p_sess->is_guest)
        {
            str_append_text(p_str, "g ");
        }
        else
        {
            str_append_text(p_str, "r ");
        }
        str_append_str(p_str, &p_sess->user_str);
    }
    str_append_char(p_str, ' ');
    /* Service name, authentication method, authentication user id */
    str_append_text(p_str, "ftp 0 * ");
    /* Completion status */
    if (succeeded)
    {
        str_append_char(p_str, 'c');
    }
    else
    {
        str_append_char(p_str, 'i');
    }
}

/* Mert SARICA */
static void
vsf_log_do_log_vsftpd_format(struct vsf_session* p_sess, struct mystr* p_str,
    int succeeded, enum EVSfLogEntryType what,
    const struct mystr* p_log_str)
{
    str_empty(p_str);
    if (!tunable_syslog_enable)
    {
        /* Date - vsf_sysutil_get_current_date updates cached time */
        str_append_text(p_str, vsf_sysutil_get_current_date());
    }
    if (!str_isempty(p_log_str))
    {
        str_append_text(p_str, " ");
        str_append_str(p_str, p_log_str);
        str_append_char(p_str, ' ');
    }
}

```

```
GNU nano 2.5.3 File: /root/honeypot/vsftpd-3.0.3/logging.c
str_append_text(p_str, "o ");
/* Access mode: anonymous/real user, and identity */
if (p_sess->is_anonymous && !p_sess->is_guest)
{
    str_append_text(p_str, "a ");
    str_append_str(p_str, &p_sess->anon_pass_str);
}
else
{
    if (p_sess->is_guest)
    {
        str_append_text(p_str, "g ");
    }
    else
    {
        str_append_text(p_str, "r ");
    }
    str_append_str(p_str, &p_sess->user_str);
}
str_append_char(p_str, ' ');
/* Service name, authentication method, authentication user id */
str_append_text(p_str, "ftp 0 * ");
/* Completion status */
if (succeeded)
{
    str_append_char(p_str, 'c');
}
else
{
    str_append_char(p_str, 'i');
}
}

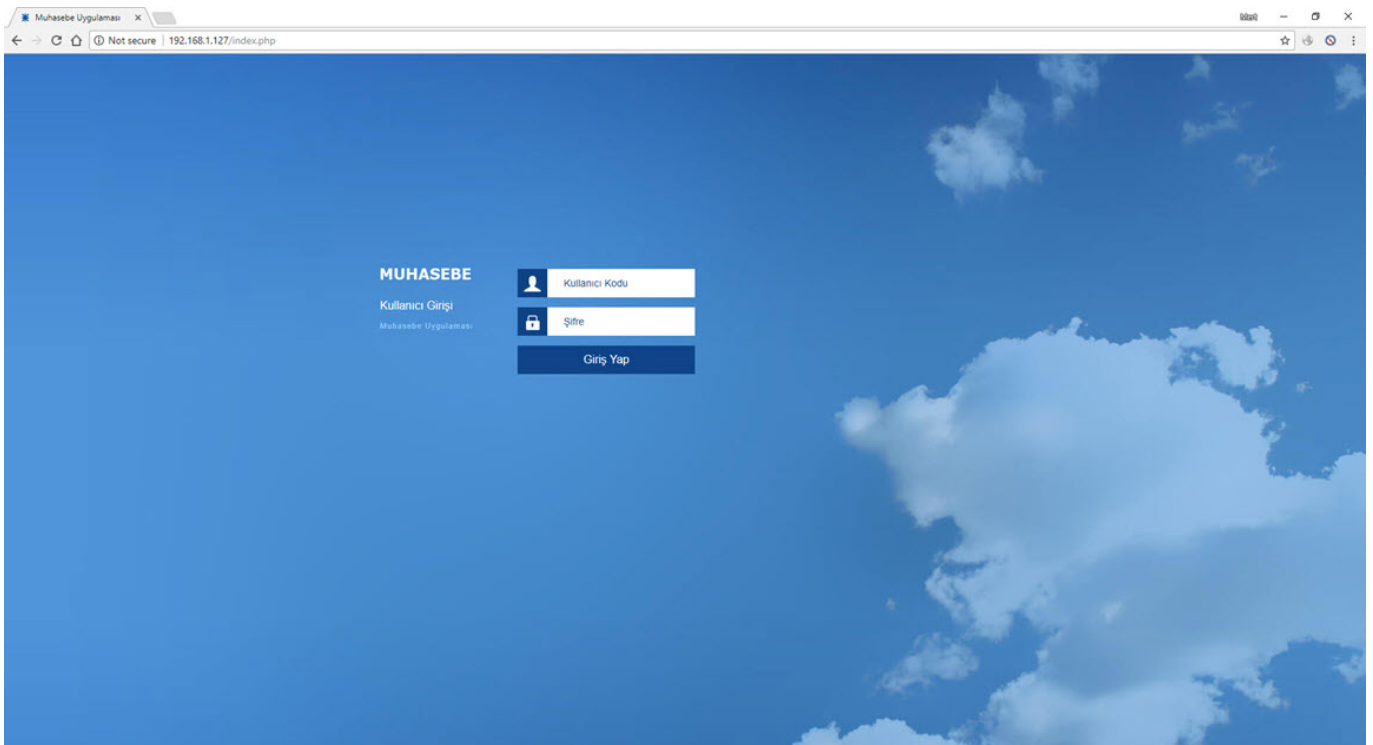
/* Mert SARICA */
static void
vsf_log_do_log_vsftpd_format(struct vsf_session* p_sess, struct mystr* p_str,
    int succeeded, enum EVSfLogEntryType what,
    const struct mystr* p_log_str)
{
    str_empty(p_str);
    if (!tunable_syslog_enable)
    {
        /* Date - vsf_sysutil_get_current_date updates cached time */
        str_append_text(p_str, vsf_sysutil_get_current_date());
    }
    if (!str_isempty(p_log_str))
    {
        str_append_text(p_str, " ");
        str_append_str(p_str, p_log_str);
        str_append_char(p_str, ' ');
    }
}

```

```
✔ Batcave | ✔ Batcave (1) | ✔ Batcave (3) | ✔ Batcave (2) x | ✔ Batcave (4)
root@ubuntu:/usr/local/pgsql/data# ftp 127.0.0.1
Connected to 127.0.0.1.
220 Muhasebe sunucusu.
Name (127.0.0.1:root): mert
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
ftp>

✔ Batcave | ✔ Batcave (1) x | ✔ Batcave (3) | ✔ Batcave (2) | ✔ Batcave (4)
root@ubuntu:/var/log# cat vsftpd.log
Sun Nov  5 17:12:15 2017: Client "127.0.0.1"
Sun Nov  5 17:12:18 2017: Client "127.0.0.1", "[Failed Password] Username: mert Password: dert"
Sun Nov  5 17:12:18 2017: Client "127.0.0.1"
root@ubuntu:/var/log#
```

Son olarak ise web sunucusunun ana klasöründe oluşturduğum index.php dosyasının, yapılan hatalı giriş denemelerini kayıt altına almasını sağladım.



```
GNU nano 2.5.3 File: index.php
}
return true;
}
</script>
<body class="ext-safari" id="ext-gen0">
<form action="index.php" method="POST" autocomplete="off" onsubmit="return (checkForm(this));">
<div class="login_panel">
  <div class="login_table">
    <div style="float:left; margin-right:20px; margin-top:3px;">
    <div style="height:30px;"></div>
    <font style="color:#FFF; font-size:16px; line-height:32px;">kullanıcı Giriş</font>
    <div style="height:6px;"></div>
    <font class="des">Muhasibe uygulaması</font>
  </div>
  <div style="float:left; margin-right:20px; margin-top:14px;">
    <div class="row" style="margin-bottom:14px;">
      <input id="username" maxlength="30" name="username" type="text" placeholder="kullanıcı Kodu" style="color:#FFF;" class="ico1">
    </div>
    <div class="row">
      <input id="password" type="password" maxlength="100" name="password" placeholder="Açık" class="ico2">
    </div>
    <div class="row">
      <input name="" id="girisbutton" type="submit" value="Giriş Yap" title="Giriş Yap">
    </div>
  </div>
</div>
</form>
<div id="ext-comp-1001" class="x-tip" style="position: absolute; z-index: 20000; visibility: hidden; display: none;"><div class="x-tip-tl"><div class="x-tip-tr"><div class="x-tip-tc"><div class="x-tip-header x
</body>
</html>
<?php
if($_REQUEST) {
  // echo $_POST['username'];
  // echo $_POST['password'];
  if(isset($_POST['username']) && isset($_POST['password'])) {
    ob_start();
    $txt = print_r($_POST, true);
    // echo $txt;
    $re1="(.*?)"; # Non-greedy match on filler
    $re2="(.*?)"; # Square Braces 1
    $re3="(.*?)"; # Any single character 1
    $re4="(.*?)"; # Square Braces 2
    $re5="(.*?)"; # Any single character 5
    if ($c=preg_match_all ("/".$re1.$re2.$re3.$re4."/is", $txt, $matches)) {
      error_log(print_r(date("d-m-Y H:i:s", $_SERVER['REQUEST_TIME']), true) . " [Failed Password] Client: " . print_r($_SERVER['REMOTE_ADDR'], true) .
        " Username: " . print_r($_POST['username'], true) . " Password: " . print_r($_POST['password'], true) . "\r\n", 3, "/var/log/apache_form.log");
    }
    ob_end_flush();
  } else {
    d.focus();
    html()turn false;
  }
} else {
  html();
}
```

```
root@ubuntu:/var/www/html# cat /var/log/apache_form.log
07-11-2017 20:40:48 [Failed Password] Client: 192.168.1.144 Username: mert Password: dert
root@ubuntu:/var/www/html#
```

Python ile geliştirdiğim loginmon isimli aracı da saat başı /var/log klasörü altında yer alan ve açık halde saklanan parolaları, hatalı giriş denemesi yapan kullanıcı adları, ip adresleri ve tarihler ile alıp, Splunk'a gönderecek şekilde hazırladım.

```

GNU nano 2.5.3 File: /etc/cron.hourly/loginmon
#!/usr/bin/python
# Failed Login Attempt Monitor v1.0
# Author: Mert Sarica
# E-mail: mert [.] sarica [ @ ] gmail [ . ] com
# URL: https://www.mertsarica.com

import time
import os
import datetime
import hashlib
import sys
import re
import smtplib
from email.header import Header
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders

debug = 0

# Log Files
logfile = "/var/log/passmon.txt"
ftptlog = "/var/log/vsftpd.log"
authlog = "/var/log/auth.log"
pglog = "/var/log/pgsql.log"
htalog = "/var/log/httpdssl.log"
formlog = "/var/log/apache_form.log"

# E-mail Parameters
sender = "mert.sarica@gmail.com"
recipient = "mert.sarica@gmail.com"
server = "127.0.0.1"
port = 25

def send_data():
    try:
        FILE = open(logfile, "rb")
    except Exception, e:
        if debug:
            return print "[*] send_data error: " + str(e)

    subject = "Failed Login Attempts"

    msg = MIMEMultipart()
    msg.set_charset("iso-8859-9")

    msg['Subject'] = str(Header(subject, "iso-8859-9"))
    msg['From'] = sender
    msg['to'] = recipient
    part = MIMEBase('application', 'octet-stream')
    part.set_payload(FILE.read())
    encoders.encode_base64(part)
    part.add_header('Content-Disposition', 'attachment; filename="passmon.txt"')

    # Attach parts into message container.
    # According to RFC 2046, the last part of a multipart message, in this case
    # the HTML message, is best and preferred.
    msg.attach(part)

    try:
        session = smtplib.SMTP(server, port)
        session.ehlo()
        session.sendmail(sender, recipient, msg.as_string())

```

```

rg = re.compile(re1+re2+re3+re4+re5+re6+re7+re8+re9+re10+re11+re12,re.IGNORECASE|re.MULTILINE)
for m in rg.finditer(txt):
    if m:
        date1=m.group(1)
        mydate = datetime.datetime.strptime(date1, '%d-%m-%Y')
        time1=m.group(2)
        sbraccess1=m.group(3)
        var1=m.group(4)
        ipaddress1=m.group(5)
        word1=m.group(6)
        var2=m.group(7)
        word2=m.group(8)
        var3=m.group(9)
        line = "Date:" + mydate.strftime('%d %b') + " Time:" + time1 + " Service:WEBFORM IP:" + ipaddress1 + " Username:" + var2 + " Password:" + var3
        print "\t[*] " + line
        log(line)

def banner():
    os.system("clear")
    print "=====
    print "Failed Login Attempt Monitor - https://www.mertsarica.com
    print "=====

def main():
    # while (1==1):
    try:
        print "[*] Analyzing logs..."
        parse_authlog()
        parse_ftptlog()
        parse_pglog()
        parse_htalog()
        parse_formlog()

    except KeyboardInterrupt:
        banner()
        print "[*] Bye..."
        sys.exit(0)

    except Exception, e:
        print "[*] Error:", str(e)
        sys.exit(1)

if __name__ == "__main__":
    banner()
    main()

```

Failed Login Attempt Monitor - https://www.mertsarica.com

```

[*] Analyzing logs...
[+] Date:7 Nov Time:20:03:07 Service:SSH IP:192.168.1.144 Username:root Password:dert
[+] Date:7 Nov Time:20:03:07 Service:SSH IP:192.168.1.144 Username:root Password:mert
[+] Date:7 Nov Time:20:03:07 Service:SSH IP:192.168.1.144 Username:root Password:test
[+] Date:5 Nov Time:17:53:23 Service:FTP IP:192.168.1.144 Username:mert Password:dert
[+] Date:9 Nov Time:22:19:35 Service:PGSQL IP:192.168.1.144 Username:root Password:mert
[+] Date:9 Nov Time:22:19:35 Service:PGSQL IP:192.168.1.144 Username:root Password:test
[+] Date:9 Nov Time:22:19:35 Service:PGSQL IP:192.168.1.144 Username:root Password:dert
[+] Date:8 Nov Time:20:16:10 Service:HTACCESS IP:192.168.1.144 Username:mert Password:dert URL:/admin/
[+] Date:8 Nov Time:20:16:12 Service:HTACCESS IP:192.168.1.144 Username:mert Password:dert URL:/admin/
[+] Date:7 Nov Time:20:40:48 Service:WEBFORM IP:192.168.1.144 Username:mert Password:dert

```



```

Batcave x
root@ubuntu:/var/log# head -n 15 passmon.txt
Date:4-12-17 Time:17:18:10 Service:SSH IP:59.63.166.83 Username:root Password:1.0
Date:4-12-17 Time:16:35:44 Service:SSH IP:59.63.166.83 Username:root Password:root!@#
Date:5-12-17 Time:19:16:21 Service:SSH IP:58.242.83.25 Username:root Password:jumpstart
Date:4-12-17 Time:16:35:48 Service:SSH IP:59.63.166.83 Username:root Password:root!@#
Date:4-12-17 Time:16:35:49 Service:SSH IP:59.63.166.83 Username:root Password:pumpkin
Date:5-12-17 Time:19:16:22 Service:SSH IP:58.242.83.25 Username:root Password:jun123
Date:6-12-17 Time:09:23:45 Service:SSH IP:42.7.26.16 Username:root Password:qweasdqwe
Date:4-12-17 Time:16:35:50 Service:SSH IP:59.63.166.83 Username:root Password:planet1
Date:6-12-17 Time:09:23:46 Service:SSH IP:42.7.26.16 Username:root Password:qweasdzxc!@#
Date:5-12-17 Time:19:16:23 Service:SSH IP:58.242.83.25 Username:root Password:jussi
Date:4-12-17 Time:16:35:50 Service:SSH IP:59.63.166.83 Username:root Password:powerpc
Date:6-12-17 Time:09:23:47 Service:SSH IP:42.7.26.16 Username:root Password:qazwsxedc!@#123
Date:5-12-17 Time:19:16:24 Service:SSH IP:58.242.83.25 Username:root Password:justdoit
Date:6-12-17 Time:09:23:48 Service:SSH IP:42.7.26.16 Username:root Password:!!@#qwe!@#
Date:5-12-17 Time:19:16:25 Service:SSH IP:58.242.83.25 Username:root Password:kaiser
root@ubuntu:/var/log#

root@ubuntu:~# /opt/splunkforwarder/bin/splunk add monitor /var/log/passmon.txt -sourcetype Loginmon -index main
Added monitor of /var/log/passmon.txt'.
root@ubuntu:~#

```

Yaklaşık 1 ay sonunda Splunk üzerinde biriken ~500.000 kayıt üzerinde gerçekleştirmiş olduğum analizde, sistemime gerçekleştirilen deneme yanılma ve sözlük saldırılarının en çok Çin'den gerçekleştirildiğini, en fazla saldırı alan servisin SSHD olduğunu, kullanıcı adı olarak en çok root, parola için ise 1234 ile deneme yapıldığını öğrenmiş oldum.

The screenshot displays the Splunk interface with a search for 'Loginmon'. The main view shows a list of events with columns for Time, Event, and various fields. A report titled 'client_country' is open, showing a bar chart of event counts by country. The top 10 values are as follows:

Top 10 Values	Count	%
China	378,896	95.429%
Portugal	6,098	1.536%
United States	2,799	0.705%
Ukraine	2,194	0.552%
France	1,835	0.462%
Singapore	780	0.196%
Italy	511	0.129%
Asia/Pacific Region	508	0.128%
India	490	0.123%
Korea, Republic of	321	0.081%

The event list below shows various SSH login attempts, including successful ones for users like 'root', 'admin', and 'www-data' from various IP addresses.

password2

>100 Values, 100% of events

Selected Yes No

Reports

Top values Top values by time Rare values

Events with this field

Top 10 Values	Count	%
1234	2,136	0.421%
admin	1,925	0.379%
root	1,453	0.286%
123456	1,107	0.218%
password	987	0.194%
support	942	0.186%
ubuntu	929	0.183%
12345	899	0.177%
raspberrypi	714	0.141%
default	709	0.14%

12/8/17 6:48:29.000 AM Date: 8-12-17 Time: 06:48:29 Service: SSH IP: 121.14.7.244 Username: admin Password: 12345
 12/8/17 6:32:36.000 AM Date: 8-12-17 Time: 06:32:36 Service: SSH IP: 123.183.209.139 Username: root Password: kokos
 12/8/17 6:32:36.000 AM Date: 8-12-17 Time: 06:32:36 Service: SSH IP: 123.183.209.139 Username: root Password: luna1234
 12/8/17 6:32:35.000 AM Date: 8-12-17 Time: 06:32:35 Service: SSH IP: 123.183.209.139 Username: root Password: iloveme
 12/8/17 6:32:04.000 AM Date: 8-12-17 Time: 06:32:04 Service: SSH IP: 123.183.209.139 Username: root Password: 123qwe!@#
 12/8/17 6:32:03.000 AM Date: 8-12-17 Time: 06:32:03 Service: SSH IP: 123.183.209.139 Username: root Password: pid
 12/8/17 6:32:03.000 AM Date: 8-12-17 Time: 06:32:03 Service: SSH IP: 123.183.209.139 Username: root Password: antinea
 12/8/17 6:31:22.000 AM Date: 8-12-17 Time: 06:31:22 Service: SSH IP: 123.183.209.139 Username: root Password: nelson
 12/8/17 6:31:22.000 AM Date: 8-12-17 Time: 06:31:22 Service: SSH IP: 123.183.209.139 Username: root Password: banban

ip

>100 Values, 100% of events

Selected Yes No

Reports

Top values Top values by time Rare values

Events with this field

Top 10 Values	Count	%
59.63.166.83	129,438	25.49%
42.7.26.15	78,242	15.40%
58.242.83.26	39,284	7.73%
61.177.172.66	39,261	7.72%
61.177.172.60	29,975	5.90%
218.87.109.154	19,342	3.80%
42.7.26.16	18,373	3.61%
61.177.172.10	14,859	2.92%
59.63.188.36	14,374	2.83%
58.242.83.33	13,912	2.74%

12/8/17 6:48:29.000 AM Date: 8-12-17 Time: 06:48:29 Service: SSH IP: 121.14.7.244 Username: admin Password: 12345
 12/8/17 6:32:36.000 AM Date: 8-12-17 Time: 06:32:36 Service: SSH IP: 123.183.209.139 Username: root Password: kokos
 12/8/17 6:32:36.000 AM Date: 8-12-17 Time: 06:32:36 Service: SSH IP: 123.183.209.139 Username: root Password: luna1234
 12/8/17 6:32:35.000 AM Date: 8-12-17 Time: 06:32:35 Service: SSH IP: 123.183.209.139 Username: root Password: iloveme
 12/8/17 6:32:04.000 AM Date: 8-12-17 Time: 06:32:04 Service: SSH IP: 123.183.209.139 Username: root Password: 123qwe!@#
 12/8/17 6:32:03.000 AM Date: 8-12-17 Time: 06:32:03 Service: SSH IP: 123.183.209.139 Username: root Password: pid
 12/8/17 6:32:03.000 AM Date: 8-12-17 Time: 06:32:03 Service: SSH IP: 123.183.209.139 Username: root Password: antinea
 12/8/17 6:31:22.000 AM Date: 8-12-17 Time: 06:31:22 Service: SSH IP: 123.183.209.139 Username: root Password: nelson
 12/8/17 6:31:22.000 AM Date: 8-12-17 Time: 06:31:22 Service: SSH IP: 123.183.209.139 Username: root Password: banban

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.