

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра Електронних обчислювальних машин

**Метод віртуалізації великих даних при аналізі
анонімних мереж**

Виконав:
студент гр. СПзм-18-2
Пугач А.В.

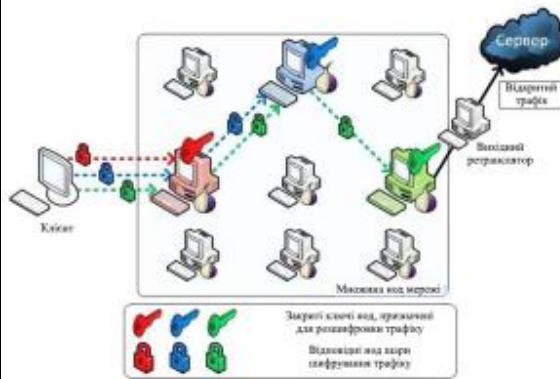
Керівник:
к.т.н., ст. викладач
Ткачов В.М.

МЕТА ВИКОНАННЯ АТЕСТАЦІЙНОЇ РОБОТИ

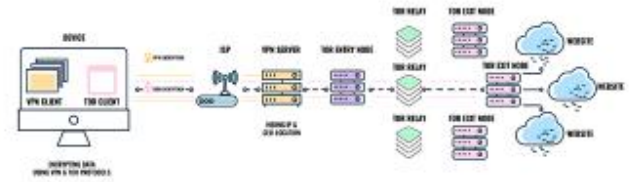
Метою даної атестаційної роботи є розробка методу віртуалізації великих даних при аналізі анонімних мереж як для вирішення задачі деанонімізації користувачів за допомогою відомих методів, так і шляхів підвищення анонімності трафіку в ненадійних сегментах мережі Інтернет.

Важливим є забезпечення швидкої обробки даних у віртуальних структурах для виявлення мережних аномалій, уражених вузлів анонімної мережі та, за необхідності, аналізу анонімної мережі з метою деанонімізації зловмисника.

МЕТОДИ ЗАБЕЗПЕЧЕННЯ АНОНІМНОСТІ В МЕРЕЖІ



Принцип роботи TOR



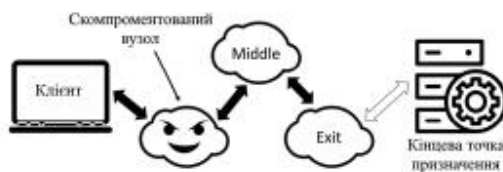
Приклад мостової схеми obfs4

3

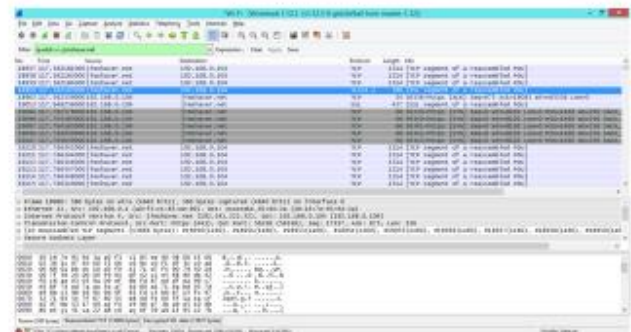
ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ АНАЛІЗУ АНОНІМНИХ МЕРЕЖ

```
c0 0a c0 14 00 39 00 38 c0 0f c0 05 00 35 c0 07
c0 09 c0 11 c0 13 00 33 00 32 c0 0c c0 0e c0 02
c0 04 00 04 00 05 00 2f c0 08 c0 12 00 16 00 13
c0 0d c0 03 fe ff 00 0a 00 ff
```

Типова сигнатура для виявлення TOR-підключень



Конфігурація ланцюга для WF атаки



Дослідження аномалій TLS трафіку

4

МЕТОД ВІРТУАЛІЗАЦІЇ ВЕЛИКИХ ДАНИХ ПРИ АНАЛІЗІ АНОНІМНИХ МЕРЕЖ

Етап 1. Вихідний список умовно розділимо на сторінки – фрагменти списку фіксованого розміру. Коли зовнішній код буде запитувати елемент колекції, вона буде завантажувати сторінку, на якій він знаходиться, і повертати потрібний її елемент. Ці дії виконує індексатор колекції, алгоритм роботи якого буде описаний нижче.

Етап 2. Крім завантаження даних, необхідно також виконувати видалення тих з них, які більше не використовуються додатком. Для цього кожна порція даних про анонімну мережу буде мати тимчасову мітку останнього звернення до неї. У певні моменти буде обчислюватися час, що минув з моменту останнього звернення до порції даних. Якщо воно перевищує якийсь порогове значення, порція даних буде видалена з колекції.

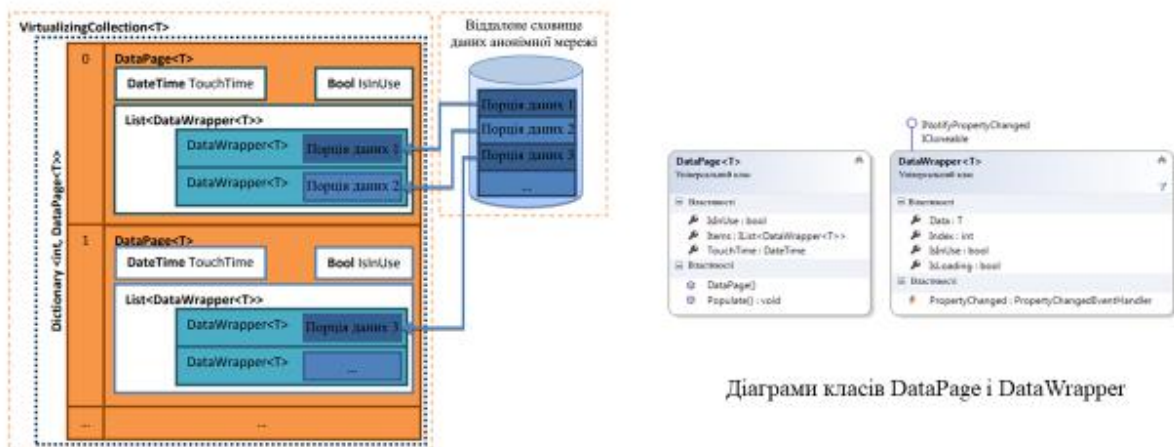
Етап 3. Оптимальне рішення для зберігання порцій даних в колекції – розміщення їх в віртуальній інфраструктурі. Цей клас реалізований як хеш-таблиця, і час виконання операцій з його елементами становить $O(1)$. Його структура дозволяє використовувати індекси порцій даних в якості ключа, що гарантує їх унікальність і спрощує пошук порцій даних. Так як крім елементів порцій даних необхідно зберігати також різну додаткову інформацію про неї, наприклад, час останнього звернення, необхідно створити клас, який представляє порцію даних колекції.

Етап 4. спосіб розміщення даних в віртуальному контейнері. Можна було б розмістити їх безпосередньо в будь-якій колекції, наприклад, в списку, і пов'язати з об'єктом порції даних. Однак такий підхід призводить до труднощів при роботі з виділенням елемента списку. Справа в тому, що анонімна мережа визначає виділений елемент на основі посилання на об'єкт, а не його індексу. Тому якщо виділений елемент, який ще не завантажений, то по завершенні завантаження виділення буде знято.

Етап 5. Використання об'єктів-обгортки, що містять об'єкти з даними як поле. Крім забезпечення коректної роботи з виділенням, обгортки дозволяють зберігати додаткову інформацію про елементи списку без змін структури класів з даними. Також вони дають можливість зробити більш досконалими механізми оновлення списку і в багатьох випадках при оновленні викликати обробники події `PropertyChanged` замість `CollectionChanged`. Подія `PropertyChanged` дає більш конкретну інформацію про зміни, що відбулися, а його обробник, відновлювальний уявлення списку, може бути викликаний з будь-якого потоку, на відміну від обробника події `CollectionChanged`, запуск якого можливий тільки з потоку графічного призначеного для користувача інтерфейсу.

5

СПОСІБ ЗБЕРІГАННЯ ДАНИХ У ВІРТУАЛЬНІЙ КОЛЕКЦІЇ

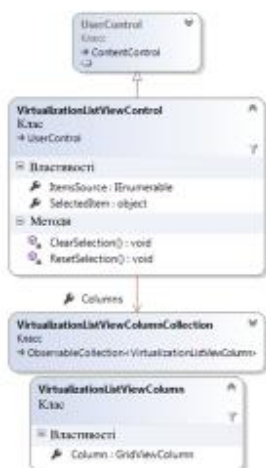


Діаграми класів `DataPage` і `DataWrapper`

Схема зберігання порцій даних для аналізу анонімної мережі в віртуальній колекції

6

ЕЛЕМЕНТ УПРАВЛІННЯ ДЛЯ РОБОТИ З ВІРТУАЛЬНОЮ КОЛЕКЦІЄЮ



Діаграма класів для
VirtualizationListViewControl

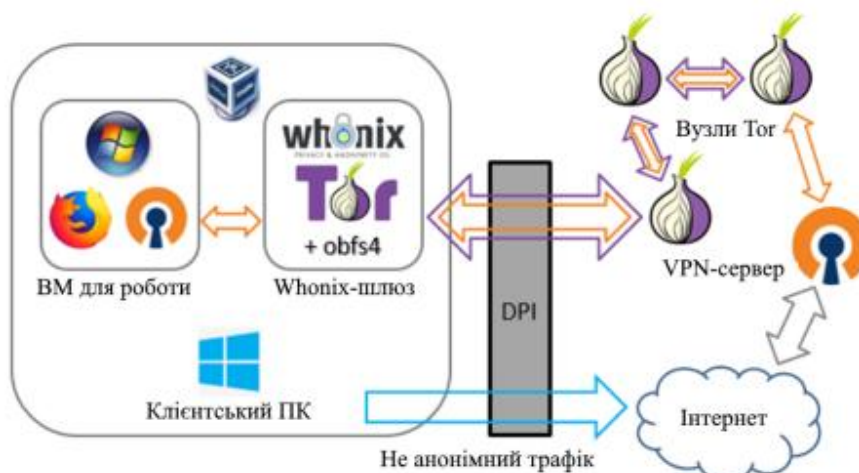
Властивість `SelectedItem` дозволяє встановлювати виділений елемент списку. Закриті методи `ClearSelection()` і `ResetSelection()` є обробниками подій зняття і відновлення виділення в списку, ініційованих віртуальною колекцією.

Дані всередині `ListView` в даному рішенні представляються в режимі `GridView` – багатостовпкового списку. Колекція стовпців для `GridView` задається через властивість `Columns` `VirtualizationListViewControl`. Є можливість задати не тільки набір стовпців, а й шаблони уявлення для даних всередині них. Також дозволяється задавати різні шаблони для об'єктів різних типів.

`ListView` налаштований таким чином, щоб він використовував всі можливі засоби підвищення продуктивності призначеного для користувача інтерфейсу. `ListView` за замовчуванням виконує віртуалізацію інтерфейсу, проте це поведінка може бути неявно відключено. Наприклад, така ситуація виникає при приміненні `ListView` всередину області прокрутки. При вирішенні задачі ситуації, що призводять до відключення віртуалізації призначеного для користувача інтерфейсу, були виключені.

Крім того, `ListView` дає можливість повторного використання контейнерів розмітки. Вона дозволяє використовувати для представлення об'єктів один і той же набір контейнерів замість того, щоб видаляти контейнери об'єктів, що вийшли з видимої області при прокручуванні, і створювати нові для введеної до неї.

ПРОЕКТУВАННЯ ПРОГРАМНОГО КОМПЛЕКСУ



Компоненти програмної системи

ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

Метод проверки	Результат	Метод проверки	Результат
Заполнен HTTP proxy	нет	Заполнен HTTP proxy	нет
Оптимиз. порта HTTP proxy	нет	Оптимиз. порта HTTP proxy	нет
Оптимиз. порта веб proxy	нет	Оптимиз. порта веб proxy	нет
Оптимиз. порта VPN	нет	Оптимиз. порта VPN	100Mbps (Pfsense)
Поддерживается изменение порта	нет	Поддерживается изменение порта	нет
Различия во времени между (Вашими и IP)	IP: 2020-05-05 07:55 (Сторонний сервер) Ваше: 2020-05-05 7:56	Различия во времени между (Вашими и IP)	IP: 2020-05-10 10:17 (ваш/Порт) Ваше: 2020-05-10 8:17
Приведенность IP к сети LAN	нет	Приведенность IP к сети LAN	нет
Режим брандмауэра Tufito	нет	Режим брандмауэра Tufito	нет
Приведенность IP к системе проксирую	нет	Приведенность IP к системе проксирую	нет
Определение веб proxy (UI метод)	нет	Определение веб proxy (UI метод)	нет
Уникал IP через Pfsense	нет	Уникал IP через Pfsense	нет
Определение уникал (дистрибуция пакет)	выполнен анализатор (на системе проксирую)	Определение уникал (дистрибуция пакет)	обфускирует
Уникал DNS	нет данных об источнике DNS	Уникал DNS	нет данных об источнике DNS
VPN bypassing	нет	VPN bypassing	MTU 1300
Уникал IP через NAT64	нет	Уникал IP через NAT64	нет

Перевірка на сайті 2ір.ua:
задовільний результат

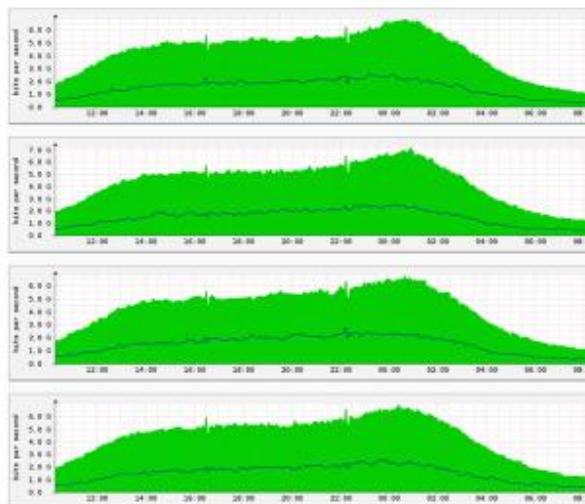
Перевірка на сайті 2ір.ua:
незадовільний результат

[illegible]

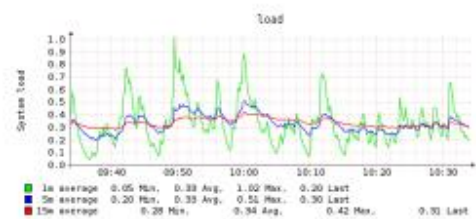
Цифровий відбиток браузера з
Panopticklick.eff.org

9

ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ



Інтенсивність надходження даних з анонімної мережі TOR на VPN-Інтерфейси серверу



Завантаженість серверу під час аналізу віртуалізованих даних з анонімної мережі

10

АПРОБАЦІЯ РЕЗУЛЬТАТІВ



Результати роботи апробовані на Міжнародній науковій інтернет-конференції "Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення" 12 травня 2020 року.

Отримана публікація:

Bondarenko M.E., Pugach A.V. Scheme of big data virtualization in anonymous computer networks // Міжнародна наукова інтернет-конференція «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення» (випуск 48), 12 травня 2020 року.

11

ВИСНОВКИ

В ході роботи над атестаційною роботою був проведений масштабний пошук різної інформації про сучасні методи ідентифікації користувачів і відстеження їх активності в Інтернеті. Це само по собі було одним із завдань даної роботи – збір і аналіз розрізнених фактів з самих різних джерел з метою систематизувати цю інформацію і перейти від неофіційного обговорення «анонімності» до наукового дослідження.

Були виконані етапи аналітичного огляду, розробки методу віртуалізації великих даних при аналізі анонімних мереж, розробки програмного забезпечення та проведення дослідницьких експериментів. В результаті була отримана конфігурація програмного комплексу, що поєднує ряд позитивних якостей. В значній мірі була підтверджена первісна гіпотеза про те, що високий рівень захисту досягається без шкоди функціональності браузера.

Спроектована програмна система має і очевидні недоліки, найбільш істотних з яких – високі системні вимоги. Віртуальна машина з Windows 7 займає значний обсяг місця на жорсткому диску і в оперативній пам'яті, при тому що браузер Firefox сам по собі споживає порівняно багато пам'яті, що призводить до необхідності виділяти віртуальній машині багато системних ресурсів, а це, відповідно, викликає брак пам'яті в основній системі. В результаті, швидкодія віртуальної машини – незадовільна. Використовуваний ланцюжок «VPN через Tor», при всіх його перевагах, може бути визнаний анонімним тільки за умови, що клієнту вдалося зберегти анонімність в результаті аналізу зібраних даних.

12

```

#!/bin/sh

rundate=$(date)

#####
# Backup the FreeNAS configuration file
#####

# Optional: specify your email address here if you want to receive
notification
email=""

# Optional: specify the short name of your ESXi host if you are running
FreeNAS
# as a VM and you want to back up the ESXi host's configuration
esxihost=""

# Specify the dataset on your system where you want the configuration
files copied.
# Don't include the trailing slash.

# Example: configdir=/mnt/tank/sysadmin/config

configdir=""

# Remove this code once you've defined configdir above... :-)

if [ -z ${configdir} ]; then
    echo "Edit script and specify the target directory ('configdir')
before using $0"
    exit 2
fi

freenashost=$(hostname -s)

fnconfigdest_version=$(< /etc/version sed -e 's/)//;s/(//;s/ /-/' | tr -
d '\n')
fnconfigdest_date=$(date +%Y%m%d%H%M%S)
fnconfigdest="${configdir}/${freenashost}-${fnconfigdest_version}"-
"${fnconfigdest_date}".db

echo "Backup FreeNAS configuration database file: ${fnconfigdest}"

iscorral=$(< /etc/version grep "Corral" | awk {'print $1'})

if [ ! -z "${iscorral}" ]; then
    # FreeNAS Corral: make a CLI call:
    cli -e "system config download path=${fnconfigdest}"
else
    # FreeNAS 9.x: Copy the source to the destination:
    cp /data/freenas-v1.db "${fnconfigdest}"
fi

l_status=$?

```



```
#####
# Backup the VMware ESXi host configuration:
#####
if [ ! -z "${esxihost}" ]; then
    esxihostname=$(ssh root@"${esxihost}" hostname)
    esxiversion=$(ssh root@"${esxihost}" uname -a | sed -e "s|VMkernel
||;s|${esxihostname}||")
    esxiconfig_url=$(ssh root@"${esxihost}" vim-cmd
hostsvc/firmware/backup_config | awk '{print $7}' | sed -e
"s|*|${esxihostname}|")
    esxiconfig_date=$(date +%Y%m%d%H%M%S)
    esxiconfig_file="${configdir}/${esxihost}-configBundle-
"${esxiconfig_date} ".tgz

    echo "Downloading $esxiconfig_url to $esxiconfig_file"
    wget --no-check-certificate --output-document="${esxiconfig_file}"
"${esxiconfig_url}"
fi
#####
# Send email notification if indicated:
#####
if [ ! -z "${email}" ]; then
    freenashostuc=$(hostname -s | tr '[:lower:]' '[:upper:]')
    freenashostname=$(hostname)
    freenasversion=$(cat /etc/version)
    logfile="/tmp/save_config.tmp"
    if [ $l_status -eq 0 ]; then
        subject="FreeNAS configuration saved on server ${freenashostuc}"
    else
        subject="FreeNAS configuration backup failed on server
${freenashostuc}"
    fi
    (
        echo "To: ${email}"
        echo "Subject: ${subject}"
        echo "Content-Type: text/html"
        echo "MIME-Version: 1.0"
        printf "\r\n"
        echo "<pre style=\"font-size:14px\">"
        if [ $l_status -eq 0 ]; then
            echo "Configuration file saved successfully on ${rundate}"
        else
            echo "Configuration backup failed with status=${l_status} on
${rundate}"
        fi
        echo ""
        echo "FreeNAS:"
        echo "Server: ${freenashostname}"
        echo "Version: ${freenasversion}"
        echo "File: ${fnconfigdest}"
        if [ ! -z "${esxihost}" ]; then
            echo ""
            echo "ESXi:"
            echo "Server: ${esxihostname}"
            echo "Version: ${esxiversion}"
            echo "File: ${esxiconfig_file}"
        fi
        echo "</pre>"
    ) > ${logfile}
    sendmail ${email} < ${logfile}
    rm ${logfile}
fi
```

```

#!/bin/sh
#
https://www.smartmontools.org/wiki/FAQ#What\_is\_error\_recovery\_control\_ERC\_and\_why\_is\_it\_important\_to\_enable\_it\_for\_the\_SATA\_disks\_in\_RAID
# ERC timeout values, in tenths of a second. The defaults below are 7
seconds for both reads and writes:
    readsetting=70
    writesetting=70
# We need a list of the SMART-enabled drives on the system. Choose one
of these
# three methods to provide the list. Comment out the two unused sections
of code.

# 1. A string constant; just key in the devices you want to report on
here:
drives="da1 da2 da3 da4 da5 da6 da7 da8 ada0"

# 2. A sysctl-based technique suggested on the FreeNAS forum:
drives=$(for drive in $(sysctl -n kern.disks); do \
    #if [ "$( /usr/local/sbin/smartctl -i /dev/${drive} | grep "SMART support
is: Enabled" | awk '{print $3}')" ]
    #then printf "${drive}" "; fi done | awk '{for (i=NF; i!=0 ; i--) print
$i }')

# 3. A smartctl-based function:
get_smart_drives()
{
    gs_drives=$( /usr/local/sbin/smartctl --scan | grep "dev" | awk '{print
$1}' | sed -e 's/\dev\/\///' | tr '\n' ' ')
    gs_smartdrives=""
    for gs_drive in $gs_drives; do
        gs_smart_flag=$( /usr/local/sbin/smartctl -i /dev/"$gs_drive" | grep
"SMART support is: Enabled" | awk '{print $4}')
        if [ "$gs_smart_flag" = "Enabled" ]; then
            gs_smartdrives="$gs_smartdrives" "${gs_drive}"
        fi
    done

    eval "$1=\$gs_smartdrives"
}
drives=""
get_smart_drives drives

# end of method 3.
set_erc()
{
    echo "Drive: /dev/$1"
    /usr/local/sbin/smartctl -q silent -l
scterc,"${readsetting}","${writesetting}" /dev/"$1"
    /usr/local/sbin/smartctl -l scterc /dev/"$1" | grep "SCT\|Write\|Read"
}
for drive in $drives; do
    set_erc "$drive"
done

```

```

#!/usr/local/bin/perl
#####
#####
#
# get-system-temps.pl
#
# Displays CPU and drive temperatures
#
# Drive information reported includes the device ID, temperature,
capacity, type
# (SDD or HDD), serial number, model, and, if available, the model
family.
#
# Optionally uses IPMI to report CPU temperatures. Otherwise, these are
pulled
# from sysctl. IPMI is more accurate in that it reports the temperature
of each
# socketed CPU in the system, even on virtualized instances, whereas the
CPU
# temperatures typically aren't available from sysctl in this case.
#
# Requires the smartmontools, available at:
https://www.smartmontools.org/
#
# Keith Nash, July 2017
#
#####
#####

use strict;
use warnings;

# Get system's hostname:

my $hostname = qx(hostname);
chomp($hostname);

# Full path to the smartctl program:

my $smartctl = "/usr/local/sbin/smartctl";

# IPMI setup:

# Toggle IPMI support on or off:
# 1 = on: use IPMI
# 0 = off: use sysctl instead of IPMI

my $useipmi = 0;

# IPMI username and password file. The password file is a text file with
the
# IPMI user's password on the first line. Be sure to set permissions to
0600
# on the password file.
#
# You may not need credentials on some systems. In this case, ignore
these
# variables and modify the ipmitool variable below to suit your
environment,
# removing the '-I lanplus' and user credential options (-U and -f) as
needed.

```

```

my $ipmiuser = "root";
my $ipmipwfile = "/root/ipmi_password";

# The IPMI host must be either an IP address or a DNS-resolvable
hostname. If you
# have multiple systems, leave the variable blank and edit the
conditional below
# to specify the IPMI host according to the host on which you are
running the script:

my $ipmihost = "";

if ($useipmi && $ipmihost eq "")
{
    if ($hostname =~ /bandit/)
    {
        $ipmihost="falcon.ipmi.spearfoot.net"
    }
    elsif ($hostname =~ /boomer/)
    {
        $ipmihost="felix.ipmi.spearfoot.net"
    }
    elsif ($hostname =~ /bacon/)
    {
        $ipmihost="fritz.ipmi.spearfoot.net"
    }
    else
    {
        die "No IPMI host specified!\n"
    }
}

# Full path to ipmitool program, including options and credentials:

my $ipmitool = "/usr/local/bin/ipmitool -I lanplus -H $ipmihost -U
$ipmiuser -f $ipmipwfile";

main();

#####
#####
#
# main
#
#####
#####
sub main
{
    printf("=====\n\n");

    if ($useipmi)
    {
        printf("%s (IPMI host: %s)\n\n", $hostname, $ipmihost);
    }
    else
    {
        printf("%s\n\n", $hostname);
    }

    display_cpu_temps();
    display_drive_info();
}

#####

```

```
#####
#
# display_cpu_temps
#
#####
sub display_cpu_temps
{
    my $temp;
    my $cpucore=0;

    if ($useipmi)
    {
        $cpucore = qx($ipmitool sdr | grep -c -i "cpu.*temp");
    }
    else
    {
        $cpucore = qx(sysctl -n hw.ncpu);
    }

    printf("=== CPU (%d) ===\n",$cpucore);

    if ($useipmi)
    {
        if ($cpucore > 1)
        {
            for (my $core=1; $core <= $cpucore; $core++)
            {
                $temp=qx($ipmitool sdr | grep -i "CPU$core Temp" | awk '{print
\ $4}');
                chomp($temp);
                printf("CPU %2u: %3sC\n",$core,$temp);
            }
        }
        else
        {
            $temp=qx($ipmitool sdr | grep -i "CPU Temp" | awk '{print \ $4}');
            chomp($temp);
            printf("CPU %2u: %3sC\n",1,$temp);
        }
    }
    else
    {
        for (my $core=0; $core < $cpucore; $core++)
        {
            $temp = qx(sysctl -n dev.cpu.$core.temperature);
            chomp($temp);
            if ($temp <= 0)
            {
                printf("CPU %2u: -N/A-\n",$core);
            }
            else
            {
                printf("CPU %2u: %3sC\n",$core,$temp);
            }
        }
    }
}

#####
#
# display_drive_info
#
```

```
#####
#####
sub display_drive_info
{
    my $drive_id;
    my $drive_model;
    my $drive_family;
    my $drive_serial;
    my $drive_capacity;
    my $drive_temp;
    my $drive_is_ssd;
    my $drive_family_display;

    printf("\n=== Drives ===\n");

    my @smart_drive_list = get_smart_drives();

    foreach my $drive (@smart_drive_list)
    {
        ($drive_model, $drive_family, $drive_serial, $drive_capacity,
$drive_temp, $drive_is_ssd) = get_drive_info($drive);

        if ($drive =~ /\dev\/(.*)/)
        {
            $drive_id = $1;
        }
        else
        {
            $drive_id = $drive;
        }

        if ($drive_family eq "")
        {
            $drive_family_display = "";
        }
        else
        {
            $drive_family_display = "(" . $drive_family . ")";
        }

        printf("%6.6s: %3uC [%8.8s %s] %-20.20s %s %s\n",
            $drive_id,
            $drive_temp,
            $drive_capacity,
            $drive_is_ssd ? "SSD" : "HDD",
            $drive_serial,
            $drive_model,
            $drive_family_display);
    }
}

#####
#####
#
# get_smart_drives
#
#####
#####
sub get_smart_drives
{
    my @retval = ();
    my @drive_list = split(" ", qx($smartctl --scan | awk '{print $1}'));

    foreach my $drive (@drive_list)
```



```

    {
        my $smart_enabled = qx($smartctl -i $drive | grep "SMART support is:
Enabled" | awk '{print \$4}');
        chomp($smart_enabled);
        if ($smart_enabled eq "Enabled")
        {
            push @retval, $drive;
        }
    }

    return @retval;
}

#####
#####
#
# get_drive_info
#
#####
#####
sub get_drive_info
{
    my $drive = shift;
    my $smart_data = qx($smartctl -a $drive);

    my $drive_model = "";
    my $drive_family = "";
    my $drive_serial = "";
    my $drive_capacity = "";
    my $drive_temp = 0;
    my $drive_is_ssd = 0;

    $drive_temp = get_drive_temp($drive);

    # Serial number
    if ($smart_data =~ /^Serial Number:\s*(.*)\s/m)
    {
        $drive_serial = $1;
    }

    # Device model
    if ($smart_data =~ /^Device Model:\s*(.*)\s/m)
    {
        $drive_model = $1;
    }

    # Model family
    if ($smart_data =~ /^Model Family:\s*(.*)\s/m)
    {
        $drive_family = $1;
    }

    # User capacity
    if ($smart_data =~ /^User Capacity:.*\[(.*)\]\s/m)
    {
        $drive_capacity = $1;
    }

    # Determine if drive is a SSD
    if ($smart_data =~ /^Rotation Rate:[ ]*Solid State Device/m)
    {
        $drive_is_ssd = 1;
    }
    elsif ($smart_data =~ /^[ 0-9]{3} Unknown_SSD_Attribute/m)

```

```

        {
            $drive_is_ssd = 1;
        }
    elif ($smart_data =~ /^[ 0-9]{3} Wear_Leveling_Count/m)
    {
        $drive_is_ssd = 1;
    }
    elif ($smart_data =~ /^[ 0-9]{3} Media_Wearout_Indicator/m)
    {
        $drive_is_ssd = 1;
    }
    elif ($drive_family =~ /SSD/)
    {
        # Model family indicates SSD
        $drive_is_ssd = 1;
    }

    return ($drive_model, $drive_family, $drive_serial, $drive_capacity,
$drive_temp, $drive_is_ssd);
}

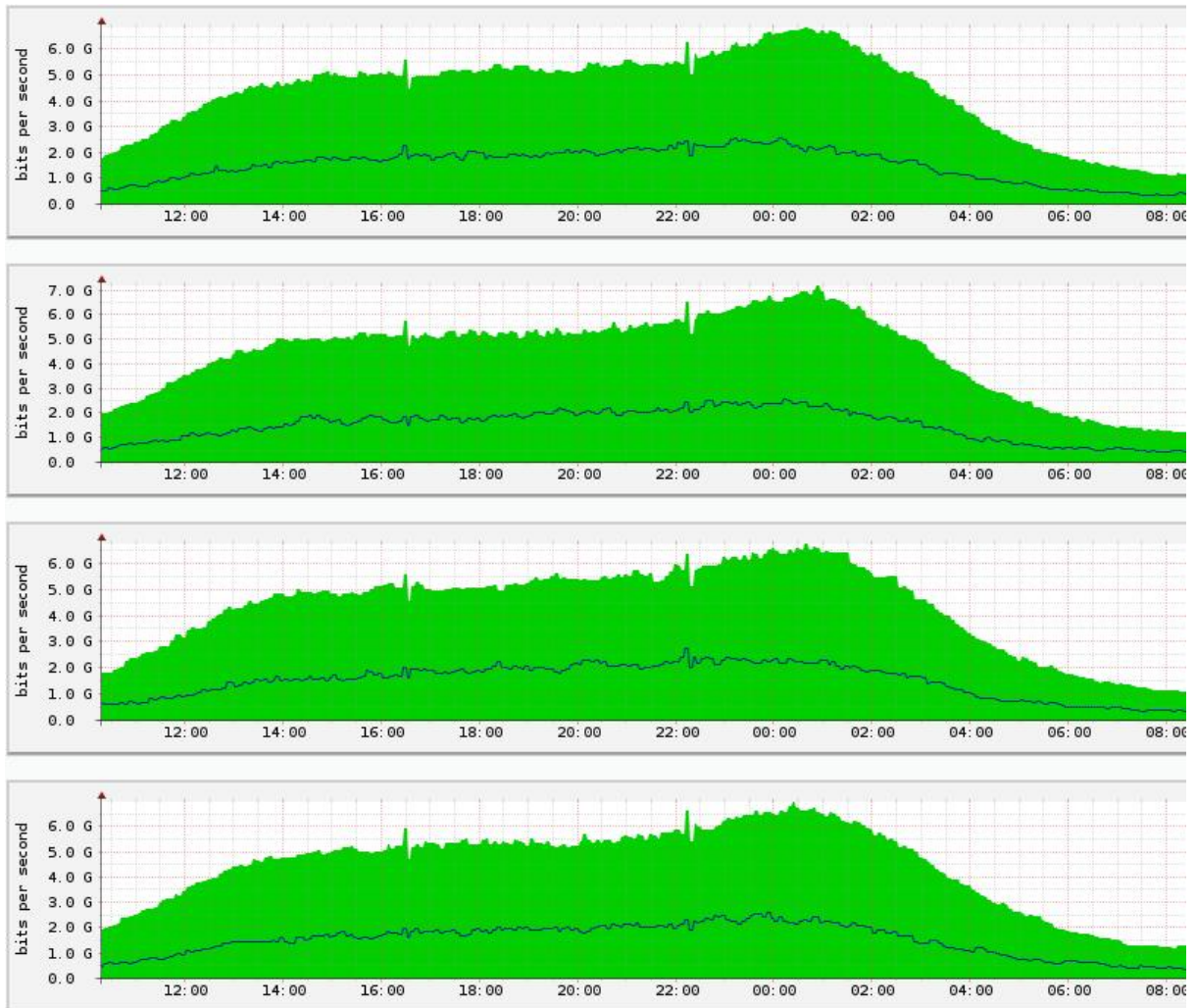
#####
#
# get_drive_temp
#
#####
sub get_drive_temp
{
    my $drive = shift;
    my $retval = 0;

    $retval = qx($smartctl -A $drive | grep "194 Temperature" | awk
'{print \$10}');

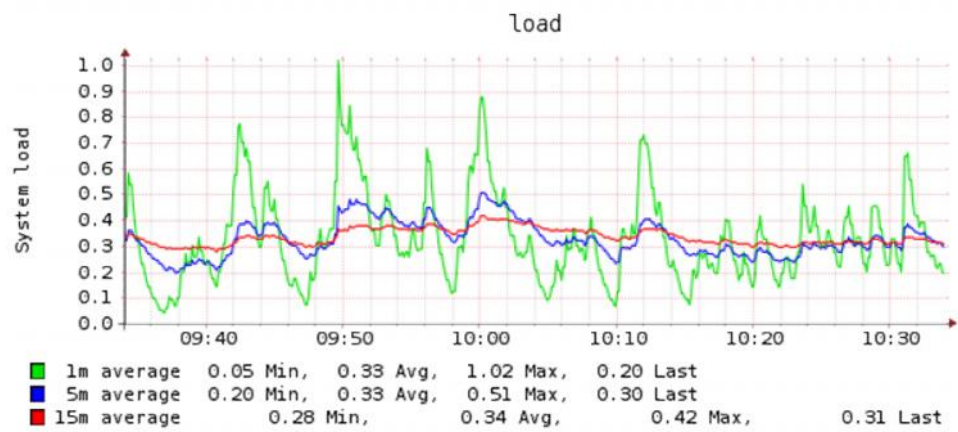
    if (!$retval)
    {
        $retval = qx($smartctl -A $drive | grep "190 Airflow_Temperature" |
awk '{print \$10}');
    }

    return $retval;
}

```



.1 –



.2 –