

① 08/11/13  
סמינר OS

מטרת הסמינר לחשוף בפנינו נושאים שאנו יודעים להם מעט  
במסגרת ההרצאה הראשונה שנה נבחרים מאמרים על נושאים  
'חמים'.

- הבהל משמחי בין מה שאנחנו ב- OS אנציואר היא מספר  
המעבדים!!! מחוברת ההפעלה צרכה לטכנן את כל הדברים  
האלה. כיום מד' ההפעלה אניצור על מעבדים שונים צברים  
צבים לתוסין. אבל אפשרים נרצה לאפליקציה אחת תלמח  
ביותר משני מעבדים ואם זה צריך אפשר...

- נושא אחר שנדבר עליו הוא אבטיה בקוצבי אב. מסתבר שאנציואר  
הזה יש תולדות שקשורה לתקשורת עם הסביבה כי חלק  
מהמכשירים הם wireless-ים. המאמר שנשחזר עליו  
מספר על פרצה (שדוד לא נוצלה) שנמצאה במכשירים  
מוביילים. באופן כללי עולם הסנסורים הוא תחום חדש  
שבה עוד נשחזר עליו הרבה.

- תחום נוסף שדוד לא בחר אמן הוא יכולת היא cloud computing.  
מד' החישוב זה משפר רחוק ואיתנו והמתלב היא משפר צר  
שיש אצלנו. יש חוג מחשבים גדול שיתלות להלכה  
שירותים (storage, זמן חישוב וכד') זה מקור  
אלטור מענינו ואולי ניעזר בה אקראי סף הסנסור.  
- האנטיבירוס היא מד' הפעלה מכוננת גדולה ויש הרבה על  
מה לדון שם...

באותו מופיעו ה צרישו המדויקו של הקורס. המטרה היא  
לא רק לזקתם אור המאמר אלא לזכור על הנושא הכללי  
ואתר ביקורו על המאמר, ולהציג עוד רעיונות. כמו כן יש  
להכין סיכום זמנה שאת הכיתה (7-5 עמודים). את  
הסיכום והמצגת יש לשלוח למרצה מראש.

נחם מנתקלים למחור נוסף ולשלוף מילן - כל הקצרים נכתבו!

# התליון: שהל (אבל הכל) יהיה מחובר ארשה.

יש הרבה חיישנים בחיי היום יום (עשן, חזרת השחר, כרכב...) הסכנות והיה ביום מאופשנה פיעה סנסוריה ממל קטנים שיש להם לא רק יכולת תישה אלא גם מקשורה אלחוטית. אבל ססור אחד הוא די מוגבל. זק מתברים הרבה סנסורים ארשה אלחוטית וביחודהם יכולים לעשה די הרבה דברים...

היתרון הגדול הוא שכרשע שיש רשע, אם סנסור אחד (פל או טעה, אצ יש את האחרים למקבים אונג. אורק החיים של הרשע הם גדול יותר כז תזק מהסנסורים יכולים אישון חזק מהמאן.

אם הסנסורים אפשר פשוט לזרוק אפה לעיך. אז הם מגיעורים ויוצרים קלטים אחד עם השן. ומתחילים זה עביר מידע לאן לעיך. אפליקציות:

- זבאור (זה גם המקור לפוויקט)
- בקרה סביבתית (אמל אפשר לשים סנסורים בעיר שמודדים טמפרטורה ומודיעים לפני שיש לפיפה)
- בית חכם (אפשר לבדוק דוב האנטרנט מה קורה בביר, אהר שקודור)
- רפואה (פיקוח על חולים בכונים - ציהוי חשש אהתקף אכ, או ציהוי רמג הסוב במל; פיקוח על חולים בבתי חולים - במקום סנסורים חוטיים; פיקוח על אולטסיה מבוגר)
- תחבורה (סנסורים על הרבים ועל הנוכחיה שרועים מידע על מצב הרבשים, מהירות מושלג, שאינה מרתק)
- זה אמור היותה אנו מוטיבציה אמה הסנסורים האלה אובים יש עדין אגדים וקלטים לא (פתרו זק הסכנות והיה לא נכנסה היום עדין. ניקלטים העיקריים הם:
- מקורה אנרגיה מוגבלים

2

- הגלגל השני יש המלאה על הליכרון ויחלפה חילוק

- א הבעיה שיכולות לצול ברשת wireless

- האפואליה של הרשת משנה בגלל א איני סיבוג (הרוח

הזיפה, סנסור האק אישון, סנסור מר) ואכן הניתוב לא

ששט א רב.

- הרשת מאבנת אר עצמה וזה שונה מרשת סטנדרטית

- בעיה scaling כי הרשת יכולה להיות גדולה מאוד (מיליונים!!!)

האתרים שצריך להתמודד איתם:

- פניסה טובה של הרשת

- הקרה על האנליטיקה - כשהצמח גבוהה יותר הקישוריות

עולה אבל אז צריך יותר אנליטיקה אר צריך אתלט על

הצפיפות של הרשת ומה בדיוק צריך

- סנסורים שהולכים אישון - מצד אחד רוצים שיטנו כפי

לשמר אנליטיקה ואורך תיים אבל זה מוריד אר הקישוריות

- לוקלליזציה - א node צריך לדעת אר המיקום שלו

כי אין הרבה משתתפים. למידע על המיקום שלו.

זאת לא GPS ? כי זה יקר, גדול, צורך הרבה אנליטיקה

אם עובד בקנינים ומתמר אקרקע.

יש שתי גישות:

• כל צול סנסורים וצד אר המיקום שלו אחד ביתם לשנ

לפי זה אפשר להסיק על הרשת

• יש nodes שונעים אר המיקום שלהם וסנסורים אחרים

אחלטיכו ביתם אליהם

- סנכרון של זמן - נעשה ע' factosync. הסנכרון נעשה רק

ושמאור צורך.

- ניתוב (צו בעיה בסיסית בלרשת) ויש פה שתי גישות

• א ה - nodes שווים ופואליים האותו אופן

• יש היררכיה ברשת (שיכולה להשתנות) אלקטורה יש אנהיג

שפיקודו אותה.

• המיקום של ה-nodes אנוני. אקביה הנתה.  
יש די הרבה פרוטוקולים אניתה להוצאו ארבוד הישגה נגדה.  
בין השאר יש פרוטוקול שמגמש בתוכן של החמלה כדי לקבות  
אח הניתה שלה (Data-centric approach)

- איחוד הניצח - נשים הרבה סנסורים באותו אזור, סמיר להניצח

לפני חלשים הוא צומח ולכן במקום לשלוח מלא הודעות כדאי

לאחד את הניצח ולשלוח רק תבולה אחת. זה גם טוב למקרה

שיש סכסוך אחד שסודה אסימה שלפי. האתגר הטו לזכור  $P$

שיגן אהראחוד האופטימלי (נובעי NP-קלה). גם אלה

יש כמה גילוי. יש אישים שביצע שזושים איחוד זה יזר

delay וזה מלחי לרובים אהימת אנו.

- אבטחה - בעיה ענקית!!! אי גשש אהשגמש הכסים

הקריפטוגרפיה הרבאים כי ינולר החילוף (מוכה, ואילו פתרון

בתחומה הם יקרים מאוד. אחת הישגה היא אגלה אבטחה

ברמה ה-link (TinySec) אהל זה פתחן אסוד קבוצת

ומיטה החומות אובדים על בעיית האבטחה.

- אסד נתנים - אנתנו רוצים שמיה יתגר אהכנים ארש שאולמ

למקבל תשובה.

נגר אדמי על החומרה ואל אע' ההפעה.

רבו קיים סנסורים לשלוח הניצח הפים והיא משגמש במעט

מאיז אנדיה. כולר האנדיה המושקעת סווח הפעולה היא 10-50 ג'

לסנסורים יש אע' הפעה TinyOS שמוכה בזיכרון, צריכה

אנדיה, ומפשה מודולריות ופיתוח. החיסרון הוא שאין

תמיכה ב-real time. היא שינה אצמי אע' ההפעה האחרונה

שאנתו מכינים (אין הפצה בין זיכרון משמש לזיכרון kernel,

אין thread-ים ועוד).

ט אפליקציה או וסריים  
 ביום חמישי / שישי  
 ואפואו טיזה ש המצגת  
 בום ראשון  
 זהבועי מראשם בני  
 צנים ארסאו אר המהל  
 הנ"ל שלו

Web Tripwires / ארצ גורן

מספר שרפי אינטרנט שמגיעים לני או  
 המהלש משתנים בזמן מהלפת. היה אתר  
 שפלה ועקבו אחרי השנויים שנשו בו  
 בזמן. (הכוונה היא לא לשנויים שנשלים  
 extensions של הדפדפן).  
 מי גאה גורם לשנויים גאה?

- ספקי אינטרנט - מכניסים פס סומור ארפים שלו בני ארציה  
 רוחים

- מנידים בכנים או מקצינים איכור של  
 מאינור כדי ארקטין או התארבורה

- ארצנים וחברור - ארצככי ארמחה ארוריו קרצ "מער"  
 - ארבריה צפים ארבינס למטמון כדי  
 ארקטין או התארבורה

- הממשל - למשל גל מיני תוכנור אחסיה פס סומור  
 או pop-ups

- פורצים - הכנסה קרצ צרוני (malware, adware)

לשנויים גאה ונרור לרור השלכור גאה צפוי

- הרפיה ונרור ארפגמ. אר מוציאים אישלו קרצ

JavaScript או מכניסים נר ינוו אמנון אר קרצ אחר

ארבור

- נר ינוו ארפגיה ארתיכה ברורומים

- פירצור מחורג בארמחה - XSS attacks . צימה

ארבה לרר הרר התוכנר Ad Muncher

פס סומור - כשאלים לרר ברורו היא מכניסה ברור

JavaScript אר שר האתר המקורי של שלו

אלו

אז אם למשל נחילי א קישור של כאורה לוקח -  
אנחנו אולי אכל למשל הכתובת היא

http://google.com/?</script>

אם השג של גוגל יתלם את התקן הא ברור ויקח את  
עמך לזיכרון, אבל בגלל Add Muncher הקובץ של  
הכל "ראה זכרון ככה:

```
<html>  
  <script>  
    //url = google.com </script>  
  <script>  
    קובץ זכרון  
  </script>  
</html>
```

לה סוגר את tag  
אפילו שנה השורה של  
הערה

וסה אמש נכרא!!!

דברים צומים אפש זכראתם השימל באתם cloak.

← המצגת / סיכום יש פירוט של הומציות שמשו והתוצאות

אז כאן תראו את הבחיה איך המצבים איתה?  
אפש כאמת זה למשל ב https - סומי בפרוטוקול  
מוצפן, אבל זה לא פתרון מושלם כי בגלל שאצפנים  
צפים, אנחנו מונעים שינויים שאנחנו כן רוצים אבצע (כמו  
proxy של חברה שבודק את ויצפים אצפנים אכאמה).  
חול אנה, זה מאט את התקשורת כי אי אפשר אשמו צפים  
במסמך וזה גם עולה יותר בגלל התאוצה לזיכרון והפיק.  
המתקן אצפן פתרון פשוט / עול יותר. הכתובת היא שהשורה  
שולה שווה אמתים אצפנים:

(4)

- הגדל

- ייצוג אמין של הגדל (עודף עומק מלא מקוצר, check sum או משפוט כזה)
- סקריפט שעושה את הבדיקה והנחשיות.

איך ממשים זאת בפרק?

- פתרון פשוט ונאיבי: פשוט סימנים או כמות ה-tags שיש בגדל ומושים השוואה עם הייצוג האמין. זה כמובן גאון נאיבי אמר המתקדמת הרבה שזה יגור מהדפוס המשיגים.
- השוואה של הייצוג המלא של הגדל. הבחירה אם זה היא שזה משתנה מצפצפן לצפצפן. איך אפשר לשלוח ושימה אמנה של check-sums אמרם בתאמה לאחד מהם אמרם אם אי אפשר לנגד את השינוי אמרם לבוא נעלה.
- XHR & overwrite
- XHR & redirect
- הפתרון שלמה מממשים בו: XHR on self
- יש איך במצגת זאת שמצגים את זה.

מבטא של השיטה האלה:

- כמו שמישהו ינוח לשנה ארוכה - מישהו ינוח גם להתעסק עם הסקריפט. אפשר לעשות כן עם איני התחכמות אמר או אפשר אמרם ארנה כ-100%
- התנגשות בין הסקריפט שבוצק אסקריפטיה שיש בגדל ואיך זה אמרם.

במצגת גרפיקה לממשים את זה - latency שנה שורם  
 עם HTTPS באופן משמעותי לוקח זמן רב יותר מ-  
 Trip Wire כי ש התהליך של ה-hand-shake  
 לוקח הרבה זמן. עם התפוקה ה-https (מכנה הרבה יותר)

בקשה זהבא : אתם סקירה יתר למלאה  
מהמאמר הספציפי. אלא סקירה של התחום  
והמאמרים הקשורים.

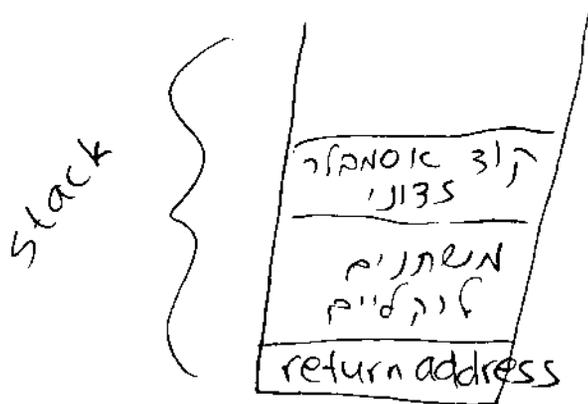
יש לנו שתיים. לא חייבים אנצל את כולן  
אבל אפשר גם אנצל יתר מהשעה הכאשרנה.

היכולת של שיטה חדשה לאבטחה נכונה הניתנת בשם WIT

מה זה פיצה אבטחה? העיור בשפה או של הקוד שמישהו צדני  
ינון לקבל שליטה על ניתונה בעלן ואדשור מה שהטו רוצה.  
זה בעיקר ינון זכרים ולק התוכנה שיש לכן אישאל המעורה-  
שיתים וטלה.

ברמה הניתון יש כמה סוגי פיצור: buffer overflow/underflow  
(אם יש תוצר הנוצר לשלו וכותבים אלו יותר מה שאתם  
וצורים כיכרון אחר); שחרור עפול (שמשתחרר אותה  
נתנה פזאיים ואז רשזשים malloc אפש לקבל פזאיים  
אותה תלבה; dangling pointers (שחרנו אה הפוינטר  
בעבר, פזאיה אה הניכרון שנוא הציבין אלו ודרישו הווא  
מציבין זמטו לא יצום).

Buffer Overflows - זוהי שכה הכי נפוצה לעשור נק.  
אפש לעשור אההתקפה הנצאג לא יק על הממסיה, אלא  
גם על ה- heap (אופה שמקצים כיכרון צינאוי)



אם יש buffer במשתנים הוקאיים אפשר לרתוב אלו  
יחי מהאוק שלו ולצרום אור נתובה הוצרה של הפוינקציה  
יק שותובה הוצרה תהיה למשל אופה שרתבנו הניכרון  
קוד אסמבלר צדוני. וזהו! אפשר להרוץ מה שבאלו!

הצורה אין אנו חתומה חכה אז אפסל למשל לזכרו  
שמו של קבצים שאחרי זה התוכנית משתמשת בהם וכו'.

### פתרונות:

Type-safe dialects - בשפות שבהן יש ניהול זיכרון  
הזכרים האלה הם קווים. לפחות כמו C ו-C++ שבהן  
יש לנו שליטה מלאה על הזיכרון הן אלא שמשפוח  
מהתקפות נאה. אז יש דיאלקטים של השפות האלה  
שהם Type-safe אבל מהמור אזיהן אה הקוד עם  
קשה.

- ספרייה בטוחה - למשל להשתמש ב-String במקום  
char\* ו-Vector במקום טרם מזיכרים.
- הגנוי- על זרכים ספציפיים כמו בתומת חזרה לה  
יזול אלה אם פתרון גלובאלי.
- הגנה על פונקציות, למשל ע"י קידוד בתקבות חזרה.  
executable space protection
- רנדומיזציה של מרחב הכתובות - משתנים לוק איימ  
במקום אחר, כתובות חזרה במקום אחר - ורגולר  
מזכרם חזרה יותר קשה ~~ל~~ לטפס אפה לזכר  
לזכרו.

- firewalls שמגנים מ- buffer overflows בזק  
הרשת.

### WIT

יתרונה: לא צריכים להתחבר למערכת או לקוד אחר  
אם סגימה - הצדקה של השפה, אמן אזי חזרה לזכרם,  
אין לו overhead גבוה.

Write-Integrity - WIT מונח מפקדות לכתיבה  
 איתור במקומות שהן לא אמורות להיות בהם. זה נעשה ע"י  
 אנטיגט - points - בודקים אם פוינטר לאילו אובייקטים  
 הוא יורה זה לביט. ואז מחזקים לבעים שלפיקודות הורחבה  
 לאובייקטים אפקודות כתיבה ינונה איתור רק לאובייקטים שהם  
 באותו זמן.

יש פקודות בטוחות - פקודות שכותבות למקום קבוע מספר זקבוע  
 של בתים. למשל  $x=y$ , אבל  $y = \text{[זיגא לאובטוחה]}$   
 $y$  לא קבוע. כל הפקודות הבאות מקבעות לבע 0 כב  
 הן לא יכולות לזפוק שום זכר.

יבטיב כאן הוא שהדייק מוכר ע"י האנליזה, שהיו האנליזה יכולה  
 להיות מאוד מסכנת.

Control-flow integrity - אמונח מחברים של התוכנית שלא לפי  
 גרף הבחינה שלה. קפיצות יאלה גם מונעים באוגה צוק - מחזקים  
 זביחה וקופצים רק לאן שמוחר. זה מונח לתדמסין קפצה של קוד  
 ש שתלנו נב ה זכר של לא מתאים לשום קפצה.  
 הוכחה הזו זחלה נב ה - write-integrity לא עובד באופן  
 משלם. אבל הוכחה הזו לא מספיקה! לנלא מאן מפני  
 buffer overflow - צימנה במציג.

אבלגור - הזכרים

מחזקים או רציונון אפליים של פ בתים ולכל אחז נותנים  
 זכר של 8 בתים. אנוני לא כוזים שהאותו חזק יתן שני  
 זכרים שונים ולכן זכק לפעמים גם padding. ה heap  
 זה לא משנה כי בלאו הני הערמה זוכרה בקוונטור של 8  
 בתים אבל של גוממטג זה יכול להשפס כי הוא מחולק -  
 אקוונטור של 4 בתים.  
 הוכחה עזמה אמונה איתורה ולק אי אפשר לשלם אותה.

• בין הצדדים של הצירוף שלהם לא בטוחים שהם מזהים שנקרא  
guards. הצדדים שלהם הוא וזה מונע את buffer overflows  
בין הצדדים האלה כי אי אפשר לבדוק את ה guards.

התוצאה הניסיונית היא באופן אישנותי יותר טובה מפתרון אחרים  
זה מהחיות זמן וזה מהחיות זיכרון (מצפים במצגת).  
זה הבדיקה בשלטי web וכיאתה תוצאה טובה.

במצגת עם סירוס של ה הבדיקה של האפקטיביות של DWT.  
מה התסכנות של DWT?

- לא מתמודדת עם קריאה ממאקרו של אסור
- תלוי בדיוק של האנליזה הסטטי-
- לא מטפל ב-slowdown בתוך אובייקטים - באור  
כל אובייקט במעט און הוא בקבוצת אחר
- צורש שינויים ברמת הקומפילציה והיא אי אפשר לערום  
עבודה למפל כמו שלצניק.

שיפויים:

- הורדת התקלה
- צמיחה של שורה שונות בתוך structures
- צמיחה שונה של הקצאה צינאמית (ולא האם באותו צבע).

8) 1/12/08  
סמינר OS

# Remus - ממשל שנותן תמיכה בפני (פול) חומרה

התארכה Remus נותנת שינוי high availability  
ומאפשרת לך virtual machine של VMX - בסופו  
של דג'יין - זו תוכנה!

ויטואליזציה זו הפרדה יותר טובה בין חומרה לתוכנה.  
זו איננה שרשרת של החומרה שנותנת ממשק אחד ומאפשרת  
לכתיב בו נתונים על החומרה כגון - גודל הפעלה, וכל זאת  
מתחת, מבחינה הממשל מנסה שהיא היחידה של כיתה.

הדג'יין הוא שמשפחה הפעלה רצה היא רוצה שבהנשלים  
של החומרה יהיו השימוש באמצעי שלה. ולכן ככה סתם או  
אפשר לכתוב כמה מערכות הפעלה בודדות. מה ש  
דושה הוא שהיא מנהלת את ההנשלים בין גודל רכוש  
ההפעלה והשנונה ואיננה זקוקה לתשובה שיהיו יחידה להצדק  
כמו שמנהלים ממשלים בין תהליכים. לה כמובן מוכיח  
קצת מה performance של ממשל הפעלה.

יש שני סוגים עיקריים של וירטואליזציה  
- full virtualization - יש תאק"ה מלאה בין הממשל  
והמארח הממשל המארח. במקרה זה, הממשל המארח  
אין מושג שיהיה VM.

para virtualization - צורה שנייה בממשל  
המארח והיא כמובן מנצחת משהיה VM.  
מתקנה זה, היי - performance נפגד בחור.

משא משתמש בגרסה השנייה של וירטואליזציה (למרות  
שהיא סופרניק שמתבטלים תוצרים הם בצלילתו עתידית חלונות  
לממשל שנייה). היכולת של משא שאיננו רצון בה

live OS migration - אחד של המצרכים מחוכה  
פיסור אתר למחנה פיסור אחת [ וזה מה ש - Remus  
אשתמש בו ] זה כמובן נעשה על ידי שהשתמש שם עם  
ונתה הצ'יטו אנוניס למה שמתן המצרכי הוא 2-60ms  
שהשתמש מהר!

ל - ה - TCP נשמרים, ו- share memory אבל  
אם הסתיימה עובדת באופן אמין.

זה שאנחנו נוצרנו חמימה הוא שאם יש איזו נפילה  
בתחילה, יתמס' בל באר מחשיבה לסיבוב. אז אפשר  
לעבור לתחילה מיותר אבל זה יקר, ואם אפשר לחתוך  
הכמה כמות האפליקציה אנו זה כמובן לא גנרי.  
המטרה של Remus היא שלא יהיה צורך לשלוח עם  
אם האפליקציה לא אר - מר' הפעלה ושהשתמש  
עם יבוא שנעשה שינוי.

הידעון בבסיסי: יש שתי מנוגות שיכור במקביל -  
primary backup: המשמש תמיד עובד עם  
ה- primary שאליו רל Remus ובתצורה  
ומאז גבוהה יוצר snapshots של ה- VM  
ומחכו אר זה - backup. נשקיות תקרה  
ה- backup ממשיך אר יוביצו והשתמש עם יוצר  
משהו יש ל- Remus מחפז (ואנחנו נדון בהק)  
אבל הוא משתמש שרם יקרה בל' תקלות ~~הוא~~ ובשקיפות  
למשמש.

סיכרון - נשנוציה לטלות משוה תחולה לניק' לטמו אר  
מנה המצרכי. וזה לא אר כי זה אר והשליטה מ-  
Remus ונתן האפליקציה לטוט. שם עם מוציאים שום  
צבני התחלה סתם ככה. יש תוצר שרם לניס' לתוכו יאתרי

9) שיוצרים checkpoint מצד מונדיטור את מה של צריך  
התוצרה. נניח שיש לנו את המלכה, ה- primary יכול  
להתחיל בחישובים שלו (ביצירה של אולי זה יירשם, אבל  
כני מסוכן כי יש גיבוי). עם זאת, גם התהליך הזה  
של שמירה ב- backup, קודם ask ורק הלאה  
סוקר SMS.

לכאורה, אחיזת כמה primary hosts מצד צדן, להכפיל  
או כמה המכונה. אבל אם זה סתמיות לתקלה נמוכה או  
אפילו להשגשוג מאותי backup צדני כמה primaries.

השטח להצטייר VM מחננה שיסיג אתה, לאחזת, צריך  
לשאת את הניכיון והיחב של ה-CPU, מספר המוצר של  
התקשורת ברשת, ועם כל הנפילות של שונות לציוד (שלא  
"יש, ע"א קונסיסטנטי) וכמובן צריך לזכור אתי "מה  
נפלה רצו שאפשר יהיה לעבור ל- backup.  
אם זה ה- primary אז זה ה- backup (נפלים)  
אז המדובר, שונה במצב קונסיסטנטי.

שני צדדים של Remus לא מסל בהם:

- תקלות בתוכנה
- אם משנו קרה לחומרה אבל היא לא יופסקה לעבוד  
(למשל, אם ה-CPU התחפרן ועושה טעויות חיסום,  
אז גם בא-ממשק ארש, אז Remus לא  
מפלים הנב).

המחזה של גרסה של ושלבים ב- live migration של  
Remus. זהו עשו להכנה שני צדדים ולפך את  
הביציות ורצי שאפשר יהיה ליצור checkpoints בתדירות  
גבוהה.

עם התקשורת אין לריק בעיה אם אילו תבולה לנולד  
לאיבוד או משנה נכה כי פלאו הכי תוכנוה אניתור להרש  
לא תמיז זובדה טיב ואם חסכה תבולה אצל תישל בקלם  
ניסטר

בדיסק הרצה נטו שונה רב אם כותבים משבול ציסק אנחנו  
וצפם שאפולו אם נפל התשמם מיז אחרי הרתיבה, תמיז  
נמצא יש, וגם אם נפל גם הציבוי אנחנו רוצים לרוב  
"ישאר בהצה קונסיסטנטי"

אז כשה primary כונה ליתובם עדיסק נאגום על עשור  
את זה, אצל מתקבול חתבים אצל זה לתוצר ה -  
backup ואז נשדשים checkpoint אצל מספורים  
אז מזה הכימון והמחבר עם backup ואז הוא עשה  
flush עדיסק שלו. אצל אם ה-primary נפל  
אז ה backup נמצא במלך קונסיסטנטי. אצל אם  
ה backup נפל באמצע ה flush אצל הוא נשאר במלך  
עם קונסיסטנטי וגם אם שניה נפלים אצל אחז מהם עם  
יהיה קונסיסטנטי אצל אי אפשר לדעת מי ...

איתור תקלה - ה backup מצפה ש כמה לנן עקבם  
מה primary check-point חזק לה - primary  
מצפה עקבם מה backup אישור שה checkpoint  
התקבם. אם עא קלה מה להם מצפים לו - סימן  
שקרה תקלה.

התוצאה של בניסויים Remus נמצאה במצגת.

# Consensus Routing

האינטרנט כמערכת מקוללת

אמשק השנים בבניית האינטרנט הייתה וזיפור לתצבותיו  
א קונסיסטנטיות יתנו לה"ג שמסודו של תבולה ישתנה  
באמצע ואם חלק מהנתבים יחלבו להמסודו תואו אחר  
ונתבים אחרים יחלבו שמקוללת הנתב אחר  
הבדלה הדיקריה תואו להתנהגות של תבולה תואו לא צפויה  
נד קורה שהישה נעלה או שיש מומס על נתב מסוים,  
ואם ~~הנתב~~ אם יש מומס קונסיסטנטיות, לא  
יבדל אם זה מסדי והמסודו תשתנה באופן חיקי, או  
שמצובה ה-abuse שלהו.

CR מתוק זשני שלבים של ניתוב תבולה:

- stable mode -

- transient mode -

כהלא באו במקום הפרוטוקול BGP הקיים, אלא מתלבש  
אליו רחוב שכבה חול משה, הפרוטוקול לא תושל בתוצה  
יתני אינפונמציה למשם אם BGP

BGP - הפרוטוקול שממשים בו היום יש טבלה של  
IP-ים שמייצאים את האפשרות לעבור בין מה-רשתות.  
הפרוטוקול שומר את המסודו של טבלה הפרוטוקול  
מוצול מאומתיוג - אם יש שינוי במסודו, נתב קוצם  
ב מחבוב את החקולה עמן לצניק ורק אח"כ מחבוב  
אם השנוי במסודו עלטר הרשתות. וזה יוצר מוסר

קונסיסטנטיות וזה יוצר בלתי התייחסות להתוארות המצב.

יש כמה פתרונות 'צדדים היום':  
- RCN - נשמית צדכון יחיד מתיים עם הסבר למה הצדכון קרה וזה עוזר להבין את התמונה הכללית.

אבל הפתרונות הידועים היום לא פותרים את כל הבעיות.

הרעיון הכללי של RC הוא להבדיל בין safety ל- liveness וזה נעשה "אלגוריתם הסכמה" - אתחלים ל- stable mode ובכך יש תקנה טובים ל- transient mode ויש שומרים את הצדכון מתוך log וכל אפיונים את הצדכון עצמו לאין הסכמה של הנתבים של הצדכון. כהנעשה באמצעות triggers.

stable mode - צדכון ה- log - עובד כמקופות שבהם תקיפה כמה לבס

distributed snapshot - בתי-רשת ליקחת תמונה של המערכת - זה אלגוריתם אנכר שיוצר מצב של מן יקוי חוצץ בין הרשתות באופן קונסיסטנטי.

frontier computation - הרשתות שלוחות את תמונת המצב שלהן אחת לשנייה ואז בעזרת אלגוריתם אפס דיברו תמונה קונסיסטנטית של הרשת.

stT computation - view change - כשנאמרת תקיפה ו+1 (ומתחילים לתקוף את +2) אפשר לזרוק את הרישומים של א.

transient mode - עומכים אלו נשמעו יציב ולא זמין יותר.  
הנרמ לעבור את ניתבולת אלפן שלו ישם לוי יש מעביר לידו עם אין לו ככה, הוא יכול לתפס אתה במסלול עק שימצא

(11)

צורך אלו לא לתמיד עובד (דוממה במצייה)

אופציה אחת היא שנושחמולה נתקלה בבציה הנתב בוחר ותב  
לא שבו ימעהי אותה אליו. הנתב החדש נופך להיוו אתראוי על  
הזכרה החמולה ליעד שלה.

עוד אפשרות היא לשמוך מסלולי גאוי ושחמולה נתקלה  
בבציה - ~~למה~~ זה לממש בהם.

ה- transient mode לה מה שקורה חק לאמשה מת"כ -  
טבלה חדשה. עדיף לנטר ערעביר אלשו נמשל לאועלות סום.

את הנופרוטיקון הזה געיקו בעזרת סמולציה - רצו לבדוק כמה  
זמן לוקח לרשם ותצור למצב יציב. התוצאות במצייה.

אבטחה

עצם זה שיש את ההפצה נה משפר את האבטחה. ממזכ  
היציב כל הנושחמור כואור את אותה תמונה עם קלפ יותר  
סתקוף. אלו אם את מהרשמו צדונוי אב אין את  
עששו.

היום נדבר על safety של דרייברים - הבטיחה והצעה אחריו. העבודה הזו היא חלק משימות של מערכת הפצה כשם Nexus שהמטרה שלה היא להיות safe - secure.

device driver הוא אמה פיסת קוד שמאפשר לנו לדבר ישירות עם החומרה. הדרייברים האלה רצים ב-kernel mode ובקדימות גבוהה. תוך מזה יש להם גישה לכל מקומות בזיכרון, עם device driver אין לנו התנה מצדו של דרייבר.

- יש פתרונות חומרה אבל הבטיחה הזיקתה שלהם נטו להם לאינדיקציה בעולם התוכנה:
- אי אפשר למנוע deadlock/live lock בדרייבר
- אי אפשר לעשות timeout לדרייבר שאינה או החוק שמן אדשוט בעולם שמו
- אי אפשר להצטרף לסייטה ספציפית - site

הפתרון המוצע הוא להפריד בין ה-devices לדרייברים ו' לרבה שמוצא את הפעולות של הדרייברים לפני שמוצאים אותם.

כמו בתוכנה המטרה היא להפגוע דרייבר ברמת הprivileges האינדיקציה הנדרשת. הדרייברים רצים ב-user mode ובכך משיג לנו safety של המערכת.

זוהי היה הסדר ארוך של אלקטרוניקה device drivers מארכיטקטורה ספציפית - 8086 של אנטל.

ההפרדה נעשה ע"י שני מנגנונים - DSS - RVM.

DSS זו בעצם מכונת מצבים. לבדוק את הפעולה שאנחנו רוצים לבצע היא חוקית. אפשר גם להיצד א-DSS שיבדוק את הקצב לבנו (אם יש) פעולה בלתי.

כל device יש DSS והוא נכתב לפי ה spec של ה device (ולאו אפי' צרכים ק"מ"מ) לאולי יש בהם בעיה בלשון). בה ערכה של הכותבים היא שיתבה של DSS אוקרת S-1 ימים במצב יש צומחה של DSS

RVM תופס את ה האירועים שקורים ב device או בדרכייה ואיוצג את זה מול ה DSS.

ברור שאנחנו רוצים ש-DSS יהי-שאל computer שלו יהי safe וצו גם הנחה סבירה כי מן שכותב אותם זה תמורה גדולה. כמובן מניחים שה RVM בטיח שה devices עצמו עובד כמו שצריך ואחרת עתק את ונצטרך אותו (אם לא). מה שאנחנו לא מניחים שיש ה- safety של הדרכייה שמול להיות שום בו באגים או שמה שישלם צדוני.

מה ה- RVM אירועים? כל מה שנמצא כמה אחר מלפניו. הוא בודק את כמח התוכנית ואולי לא וצו גם צריכה לא חושב את מלוא הימחה של ה device.

כאן היה הסדר על הממש של הנמצא מלך Nexus.

במצב יש כמה שמסמך את כמח ויקוד עתיה צריך ארוס / אשור בקוד של הדרכייה באינקה שלהם בדקו הבניה הפעולה: במערכת מאוסה הפרכייה לא וצו

(13)

כמו שציינת.

הבדיקה (ע"ש) על מחשבים ביתיים היא כמובן . התוצאה כמובן .

השוואה בין ספקי אינטרנט www.netdiff.org

מה זה ISP? יש לומר את המילים שלצדק זהותם עליהם  
- ספקי הולכה (כאלו זה הכפלים :- ADSL)

מה צדק לבחון את הביצועים של ספקי? איתנו נתקד  
בדיוק ב- latency וזה משפס על הרבה צברים:  
- אקונומי ו אפליקציו

- בחירת ספקי האינטרנט הרצויה  
- אפליקציות מבוצרות שתלויה מאוד ברשט שהן עוברות בה  
- אנולו זהה אין מספיק משאג זה היום

השיטות הק"מור כיום:

- SLA - הספקי באה ואומרת מה היא מבטיחה  
למיד השטח שלה. באבד. כל מה שקורה בחוץ הוא לכאורה  
לא באחריותה אבל. איתנו נראה שנוקא יל השפעה.  
- מאת מצטרף אפש לבדוק אם מה שהבטיחו לנו זה אכן  
מה לקיבלנו - keynote. בדעיה עם keynote הוא  
שלה, היא נותנת מיצט כק יצגי מה שקורה בתוך השטח  
של הספקי.

המטרה של Netdiff

- לאפיין את האיותה של ה traffic עבור מסלול  
אסוי  
- א מדוד א ארן קוויפונטיו- פנימיוה אא אה כל ה מסלול כולו  
- שה השוואה תהיה הולגת (כמו שהספקי יצדווה)

ה. latency צננה יותר וטוב  
- אצטונז אסקור אהבין מה כצאא לאלפס

### Netdiff בארויקטורה של

המזכה האויבאי הוא שיש (קצרה מצידה הכו POP אהכל  
רשג מטרה. אבל זה מכנס הרבה overhead  
- keynode יש (קצרה מצידה הק כמה) POPs של  
ISPs שמשפטים פעולה. והמזכה שלהם יצבה איותמה  
קוצאור ורנות אבל כמו שכבר אמרנו, המזכה הוא  
לא גלובליים.

צתונה זשנהת הוקצור קצה של הרשג (מתלבים ביותיים  
סוגל) ועי התלפה (תנוים בין. (קצרה שנוג מתלבים  
אר התנוצאור. עכן צריק. אהוקיס אנוצ אר כנוור  
המצידה לעושים כי מתשכ ביותו לא יורו אהתמוצד עם  
מצידה בוצים של (הרשג) - א250 מטרו שנוג.

השג המציה ואסיכום